NetScaler 11.1

Contents

NetScaler Release Notes **3**

Getting started with Citrix ADC **4**

Where does a NetScaler appliance fit in the network? **7**

How a NetScaler appliance communicates with clients and servers **10**

Introduction to the NetScaler product line **17**

Install the hardware **18**

Access a NetScaler appliance **19**

Configure the ADC for the first time **23**

Secure your NetScaler deployment **24**

Configure high availability **24**

Change an RPC node password **29**

Configure a FIPS appliance for the first time **30**

Common network topologies **33**

System management settings **38**

System settings **38**

Packet forwarding modes **40**

Network interfaces **47**

Clock synchronization **48**

DNS configuration **49**

SNMP configuration **51**

Verify configuration **55**

Load balance traffic on a NetScaler appliance **58**

Load balancing **59**

Persistence settings **63**

Configure features to protect the load balancing configuration **69**

A typical load balancing scenario **72**

Accelerate load balanced traffic by using compression **76**

Secure load balanced traffic by using SSL **83**

Features at a glance **101**

Application switching and traffic management features **101**

Application acceleration features **106**

Application security and firewall features **107**

Application visibility feature **109**

Solutions for Telecom Service Providers **111**

Large Scale NAT **112**

Points to Consider before Configuring LSN **117**

Configuration Steps for LSN **119**

Sample LSN Configurations **137**

Configuring Static LSN Maps **147**

Configuring Application Layer Gateways **150**

Application Layer Gateway for FTP, ICMP, and TFTP Protocols **151**

Application Layer Gateway for PPTP Protocol **153**

Application Layer Gateway for SIP Protocol **155**

Application Layer Gateway for RTSP Protocol **169**

Logging and Monitoring LSN **174**

STUN Timeout **199**

TCP SYN Idle Timeout **201**

Overriding LSN configuration with Load Balancing Configuration **202**

Clearing LSN Sessions **204**

Load Balancing SYSLOG Servers **206**

Port Control Protocol **208**

Dual-Stack Lite **211**

Points to Consider before Configuring DS-Lite **215**

Configuring DS-Lite **216**

Configuring DS-Lite Static Maps **226**

Configuring Deterministic NAT Allocation for DS-Lite **228**

Configuring Application Layer Gateways for DS-Lite **230**

Application Layer Gateway for FTP, ICMP, and TFTP Protocols **231**

Application Layer Gateway for SIP Protocol **231**

Application Layer Gateway for RTSP Protocol **234**

Logging and Monitoring DS-Lite **236**

Port Control Protocol for DS-Lite **245**

Large Scale NAT64 **247**

Points to Consider for Configuring Large Scale NAT64 **252**

Configuring DNS64 **253**

Configuring Large Scaler NAT64 **255**

Configuring Application Layer Gateways for Large Scale NAT64 **261**

Application Layer Gateway for FTP, ICMP, and TFTP Protocols **261**

Application Layer Gateway for SIP Protocol **261**

Application Layer Gateway for RTSP Protocol **264**

Configuring Static Large Scale NAT64 Maps **266**

Logging and Monitoring Large Scale NAT64 268

Port Control Protocol for Large Scale NAT64 280

Telco subscriber management 282

Subscriber aware traffic steering 299

Subscriber aware service chaining 304

Policy based TCP profile selection 311

Load Balance Control-Plane Traffic that is based on Diameter, SIP, and SMPP Protocols 313

Provide DNS Infrastructure/Traffic Services, such as, Load Balancing, Caching, and Logging for Telecom Service Providers 314

Provide Subscriber Load Distribution Using GSLB Across Core-Networks of a Telecom Service Provider 315

Bandwidth Utilization Using Cache Redirection Functionality 315

NetScaler TCP Optimization 316

Getting Started 316

Management Network 319

Licensing 319

High Availability 320

Gi-LAN Integration 321

TCP optimization configuration 327

Analytics and Reporting 333

Real-time Statistics 333

SNMP 335

Technical Recipes 338

Scalability 341

Optimizing TCP Performance using TCP Nile 347

Troubleshooting Guidelines **357**

Frequently Asked Questions **359**

NetScaler Solutions **363**

Setting up NetScaler for Citrix Virtual Apps and Desktops **363**

Global Server Load Balancing (GSLB) Powered Zone Preference **365**

NetScaler in a Private Cloud Managed by Microsoft Windows Azure Pack and Cisco ACI **366**

Creating a NetScaler Load Balancer in a Plan in the Service Management Portal (Admin Portal) **368**

Configuring a NetScaler Load Balancer by Using the Service Management Portal (Tenant Portal) **370**

Deleting a NetScaler Load Balancer from the Network **376**

Deploying Citrix NetScaler VPX **378**

Support matrix and usage guidelines **378**

Install a NetScaler VPX instance on XenServer **383**

Configure VPX instances to use single root I/O virtualization (SR-IOV) network interfaces **386**

Install a NetScaler VPX instance on VMware ESX **388**

Configure a NetScaler VPX instance to use VMXNET3 network interface **393**

Configure a NetScaler VPX instance to use SR-IOV network interface **405**

Migrating the NetScaler VPX from E1000 to SR-IOV or VMXNET3 Network Interfaces **423**

Configure a NetScaler VPX instance to use PCI passthrough network interface **424**

Install a NetScaler VPX instance on Microsoft Hyper-V server **427**

Install a NetScaler VPX instance on Linux-KVM platform **432**

Prerequisites for installing a NetScaler VPX instance on Linux-KVM platform **433**

Provision the NetScaler VPX instance by using OpenStack **438**

Provisioning the NetScaler Virtual Appliance by using the Virtual Machine Manager **447**

Configure a NetScaler VPX instance to use SR-IOV network interfaces **455**

Configure a NetScaler VPX instance to use PCI passthrough network interfaces **465**

Provision the NetScaler VPX instance by using the virsh program **469**

Manage the NetScaler VPX guest VMs **473**

Deploy a NetScaler VPX instance on AWS **476**

Limitations and usage guidelines **480**

Deploy a NetScaler VPX standalone instance on AWS **482**

Download a NetScaler VPX license **487**

Load balancing servers in different availability zones **487**

Deploy a high availability pair on AWS **488**

Upgrade a NetScaler VPX instance on AWS **495**

Troubleshoot a VPX instance on AWS **500**

Deploying Citrix NetScaler VPX on Microsoft Azure **500**

Configure a NetScaler VPX standalone instance **509**

Configure multiple IP addresses for a NetScaler VPX standalone instance **521**

Configure GSLB on NetScaler VPX instances **528**

Configure address pools (IIP) for a Citrix Gateway appliance **536**

Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands **538**

PowerShell scripts for Azure deployment **546**

Azure terminology **560**

Deploy a NetScaler VPX instance on Oracle Public Cloud **563**

Prerequisites **565**

Configure a NetScaler VPX standalone instance on Oracle Public Cloud **566**

Scenario **570**

Deploy a NetScaler VPX Instance on Cisco CSP 2100 **578**

Jumbo frames on NetScaler VPX instances **580**

Hardware Installation **582**

Licensing **582**

NetScaler licensing overview **583**

NetScaler Pooled Capacity **590**

Upgrade and downgrade a NetScaler appliance **591**

Before you begin **591**

Upgrade considerations - SNMP configuration **592**

Upgrade a Citrix NetScaler standalone appliance **594**

Downgrade a Citrix NetScaler standalone appliance **600**

Upgrade a high availability pair **604**

Downgrade a high availability pair **610**

Troubleshooting **610**

FAQs **615**

New and Deprecated Commands, Parameters, and SNMP OIDs **615**

Deprecated features and functionalities **632**

AAA application traffic **634**

How AAA works **634**

Enabling AAA **637**

Setting up an authentication virtual server **638**

Configuring the authentication virtual server **640**

Configuring a traffic management virtual server **643**

Configuring DNS **645**

Verifying your setup for AAA **645**

Creating an authentication profile **646**

Configuring users and groups **647**

Configuring AAA policies **650**

Authentication policies **651**

LDAP authentication policies **656**

Negotiate authentication policies **658**

RADIUS authentication policies **659**

SAML authentication policies **663**

Web authentication policies **663**

Configuring advanced authentication policies **665**

Authorizing user access to application resources **666**

Auditing authenticated sessions **667**

Session settings **669**

Session profiles **670**

Session policies **672**

Global session settings **676**

Traffic Settings **677**

Traffic profiles **677**

Traffic policies **679**

Form SSO profiles **681**

SAML SSO profiles **683**

Session timeout for OWA 2010 **685**

Authenticating with client certificates **686**

Client certificate pass-through **688**

Configuring AAA with commonly used protocols **690**

Handling authentication, authorization and auditing with Kerberos/NTLM **690**

How NetScaler appliance implements Kerberos for client authentication **692**

Configuring kerberos authentication on the NetScaler appliance **695**

Configuring kerberos authentication on a client **698**

Offloading Kerberos authentication from physical servers **699**

NetScaler appliance kerberos single sign-on **703**

An overview of NetScaler appliance kerberos SSO **703**

Setting up NetScaler appliance SSO **706**

Prerequisites **706**

Configuring SSO **711**

Enabling integrated authentication on the web application server **711**

Setting up SSO by impersonation **712**

Configuring SSO by Delegation **713**

Generating the KCD keytab script **719**

SAML authentication **720**

NetScaler appliance as a SAML SP **722**

NetScaler appliance as a SAML IdP **725**

Configuring SAML single sign-on **728**

OAuth authentication **729**

Multi-Factor (nFactor) authentication **731**

How to configure nFactor authentication **732**

How nFactor authentication works **733**

Configuring nFactor authentication **734**

Configuration support for SameSite cookie attribute **736**

Admin Partitioning **739**

Supported NetScaler configurations **746**

Configure admin partitions **751**

VLAN configuration for admin partitions **760**

SNMP support for admin partitions **768**

AppExpert **770**

Action analytics **771**

Configure a selector **772**

Configure a stream identifier **774**

View statistics **776**

Grouping records on attribute values **779**

Clearing a stream session **782**

Configure policy for optimizing traffic **783**

How to limit bandwidth consumption for user or client device **784**

AppExpert applications and templates **787**

How AppExpert Application Works **789**

Get started with AppExpert **790**

Downloading an application template **791**

Importing an Application Template **791**

Verifying and Testing Application Configuration **792**

Customizing the configuration **794**

Configure public endpoints **794**

Configure services and service groups for an application unit **795**

Create application units **796**

Configuring application unit rules **796**

Configuring policies for application units **797**

Configuring Application Units **802**

Configuring public endpoints for an application **803**

Specifying the Order of Evaluation of Application Units **804**

Configuring Persistency Groups for Application Units **805**

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer **806**

Configuring User Authentication, Authorization, and Auditing **806**

Monitoring a NetScaler Application **807**

Deleting an Application **808**

Configure application authentication, authorization, and auditing **808**

Setting Up a Custom NetScaler Application **811**

Creating and managing template files **815**

Exporting an appExpert application to a template file **815**

Exporting a Content Switching Virtual Server Configuration to a Template File **816**

Creating Variables in Application Templates **818**

Uploading and downloading template files **819**

Understanding NetScaler application templates and deployment files **820**

Deleting a Template File **825**

NetScaler Gateway Applications **825**

Adding Intranet Subnets **827**

Adding Other Resources 828

Configuring Authorization Policies 828

Configuring Traffic Policies 829

Configuring Clientless Access Policies 830

Configuring TCP Compression Policies 831

Configure bookmarks 832

Customizing the configuration 832

Configure public endpoints 833

Configuring Endpoints for an Application Unit 833

Configure services and service groups for an application unit 834

Configuring services, service groups, and load balancing Parameters for an application unit 835

Create application units 836

Configuring application unit rules 836

Configuring policies for application units 837

Create application units 842

Configuring public endpoints for an application 843

Specifying the Order of Evaluation of Application Units 843

Configuring Persistency Groups for Application Units 844

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer 845

AppQoE 845

Enabling AppQoE 847

AppQOE actions 847

AppQoE parameters 852

AppQoE Policies 853

Entity templates **855**

Configuring an entity template **856**

Configuring variables in load balancing virtual server templates **857**

Modifying an entity template **859**

Deleting an entity template **860**

Creating an entity from a template **860**

Managing entity template folders **862**

Uploading and downloading entity templates **862**

Understanding load balancing entity templates and deployment files **863**

HTTP callouts **866**

How an HTTP callout works **867**

Notes on the format of HTTP requests and responses **868**

Configuring an HTTP callout **870**

Verifying the Configuration **873**

Invoking an HTTP Callout **874**

Avoiding HTTP callout recursion **876**

Caching HTTP Callout Responses **878**

Use Case: Filtering Clients by Using an IP Blacklist **879**

Use Case: ESI Support for Fetching and Updating Content Dynamically **881**

Use Case: Access control and authentication **884**

Use Case: OWA-Based Spam Filtering **886**

Use Case: Dynamic Content Switching **890**

Pattern sets and data sets **891**

How string matching works with pattern sets and data sets **892**

Configuring a Pattern Set **893**

Configuring a data set **895**

Using pattern sets and data sets **896**

Sample usage **896**

Variables **899**

Configuring and using Variables **900**

Use Case: Caching User Privileges **903**

Use Case: Limiting the Number of Sessions **904**

Policies and expressions **906**

Introduction to policies and expressions **911**

Classic and advanced policies **911**

Classic and advanced policy expressions **920**

Converting policy expressions using NSPEPI tool **922**

Before you proceed **926**

Configure advanced policy infrastructure **926**

Rules for names in identifiers used in policies **927**

Create or modify a policy **928**

Policy configuration examples **930**

Bind policies using advanced policy **931**

Bind a policy globally **939**

Bind a policy to a virtual server **943**

Display policy bindings **945**

Unbind a policy **946**

Create policy labels **949**

Configure a policy label or virtual server policy bank 953

Invoke or remove a policy label or virtual server policy bank 960

Configure and bind policies with the policy manager 965

Configuring advanced policy expression: getting started 968

Basic elements of an advanced policy expression 968

Compound advanced policy expressions 973

Specify the character set in expressions 986

Classic expressions in advanced policy expressions 989

Configure advanced policy expressions in a policy 990

Configure named advanced policy expressions 994

Configure advanced policy expressions outside the context of a policy 996

Advanced policy expressions: evaluating text 997

About text expressions 998

Expression prefixes for text in HTTP requests and responses 1000

Expression prefixes for VPNs and clientless VPNs 1001

Basic operations on text 1010

Complex operations on text 1015

Advanced policy expressions: working with dates, times, and numbers 1029

Format of dates and times in an expression 1029

Expressions for the NetScaler system time 1030

Expressions for SSL certificate dates 1034

Expressions for HTTP request and response dates 1042

Generate the day of the week, as a string, in short and long formats 1043

Expression prefixes for numeric data other than date and time 1044

Converting numbers to text **1045**

Virtual server based expressions **1046**

Advanced policy expressions: Parsing HTTP, TCP, and UDP data **1048**

About evaluating HTTP and TCP payload **1048**

Expressions for identifying the protocol in an incoming IP packet **1049**

Expressions for HTTP and cache-control headers **1051**

Expressions for extracting segments of URLs **1071**

Expressions for HTTP status codes and numeric HTTP payload data other than dates **1072**

SIP expressions **1073**

Operations for HTTP, HTML, and XML encoding and "safe" characters **1084**

Expressions for TCP, UDP, and VLAN data **1087**

Expressions for evaluating a DNS message and identifying its carrier protocol **1091**

XPath and HTML, XML, or JSON expressions **1093**

Encrypt and decrypt XML payloads **1096**

Advanced policy expressions: parsing SSL **1099**

Advanced policy expressions: IP and MAC addresses, throughput, VLAN IDs **1103**

Advanced policy expressions: stream analytics functions **1110**

Advanced policy expressions: DataStream **1111**

Typecasting data **1124**

Regular Expressions **1124**

Basic characteristics of regular expressions **1125**

Operations for regular expressions **1126**

Configuring classic policies and expressions **1128**

Configure a classic policy **1129**

Configure a classic expression **1131**

Bind a classic policy **1134**

View classic policies **1137**

Create named classic expressions **1138**

Expressions reference-advanced policy expressions **1140**

Expressions reference-classic expressions **1140**

Summary examples of default syntax expressions and policies **1151**

Tutorial examples of default syntax policies for rewrite **1159**

Tutorial Examples of Classic Policies **1163**

Migrate Apache mod_rewrite rules to default policy syntax **1169**

Rate limiting **1184**

Configuring a Stream Selector **1185**

Configuring a traffic rate limit identifier **1186**

Configuring and Binding a Traffic Rate Policy **1188**

Viewing the Traffic Rate **1189**

Testing a Rate-Based Policy **1190**

Examples of Rate-Based Policies **1191**

Sample Use Cases for Rate-Based Policies **1193**

Rate Limiting for Traffic Domains **1195**

Configure rate limit at packet level **1196**

Responder **1197**

Enabling the Responder Feature **1199**

Configuring a Responder Action **1200**

Configuring a Responder Policy **1204**

Binding a Responder Policy **1206**

Setting the default action for a responder policy **1208**

Responder action and policy examples **1210**

Diameter Support for Responder **1212**

RADIUS Support for Responder **1214**

DNS Support for the Responder Feature **1218**

Troubleshooting **1220**

Rewrite **1221**

How rewrite works **1223**

Enabling the rewrite feature **1226**

Configuring a Rewrite Action **1227**

Configuring a Rewrite Policy **1232**

Binding a Rewrite Policy **1235**

Configuring Rewrite Policy Labels **1239**

Configuring the Default Rewrite Action **1240**

Bypassing the Safety Check **1242**

Rewrite Action and Policy Examples **1243**

Example 1: Delete Old X-Forwarded-For and Client-IP Headers **1244**

Example 2: Adding a Local Client-IP Header **1246**

Example 3: Tagging Secure and Insecure Connections **1247**

Example 4: Mask the HTTP Server Type **1248**

Example 5: Redirect an External URL to an Internal URL **1249**

Example 6: Migrating Apache Rewrite Module Rules **1250**

Example 7: Marketing Keyword Redirection **1251**

Example 8: Redirect Queries to the Queried Server **1252**

Example 9: Home Page Redirection **1253**

URL Transformation **1254**

Configuring URL Transformation Profiles **1255**

Configuring URL Transformation Policies **1259**

Globally Binding URL Transformation Policies **1262**

Globally Binding URL Transformation Policies **1264**

Diameter Support for Rewrite **1266**

DNS Support for the Rewrite Feature **1268**

String maps **1270**

URL Sets **1272**

Getting Started **1273**

Advanced Policy Expressions for URL Evaluation **1274**

Configuring URL Set **1275**

URL Pattern Semantics **1278**

URL Categories **1279**

AppFlow **1285**

Configuring the AppFlow Feature **1289**

Exporting Performance Data of Web Pages to AppFlow Collector **1297**

Session reliability on NetScaler high availability pair **1300**

Application Firewall **1302**

FAQs and Deployment Guide **1306**

Introduction **1315**

Web application security **1315**

How App Firewall works **1319**

The App Firewall configuration interfaces **1322**

Configuring the App Firewall **1323**

Enabling App Firewall **1327**

The App Firewall wizard **1328**

Manual configuration **1334**

Manual configuration by using the GUI **1335**

Manual configuration By using the command line interface **1346**

Signatures **1348**

Manually configuring the signatures feature **1352**

Adding or removing a signatures object **1352**

Configuring or modifying a signatures object **1354**

Protecting JSON applications using signatures **1358**

Updating a signature object **1362**

Exporting a signatures object to a file **1365**

The Signatures editor **1365**

To add a signature rule category **1367**

Signature rule patterns **1368**

To Import and merge rules **1373**

Signature updates in high availability deployment and build upgrades **1374**

Overview of security checks **1375**

Top level protections **1377**

HTML cross-site scripting check **1378**

HTML SQL injection check **1389**

Buffer overflow check **1403**

Cookie consistency check **1409**

App Firewall support for Google web toolkit **1413**

Data leak prevention checks **1418**

Credit card check **1418**

Safe object check **1426**

Advanced form protection checks **1428**

Field formats check **1429**

Form field consistency check **1443**

CSRF form tagging check **1445**

Managing CSRF form tagging check relaxations **1448**

URL protection checks **1449**

Start URL check **1449**

Deny URL check **1454**

XML protection checks **1455**

XML format check **1455**

XML denial-of-service check **1456**

XML cross-site scripting check **1459**

XML SQL injection check **1466**

XML attachment check **1475**

Web services interoperability check **1476**

XML message validation check **1480**

XML SOAP fault filtering check **1481**

Managing content types **1482**

Profiles **1486**

Creating App Firewall profiles **1488**

Configuring App Firewall profiles **1490**

Changing an App Firewall profile type **1492**

Exporting and importing an App Firewall profile **1493**

Configuring and using the Learning feature **1498**

Supplemental Information about profiles **1504**

Policy labels **1510**

Policies **1512**

App Firewall Policies **1512**

Creating and configuring App Firewall policies **1514**

Binding App Firewall policies **1519**

Viewing a policy bindings **1521**

Supplemental information about App Firewall policies **1521**

Auditing policies **1522**

Imports **1527**

Importing and exporting files **1530**

Global configuration **1532**

Engine settings **1533**

Confidential fields **1536**

Field types **1541**

XML content types **1543**

JSON content types **1545**

Statistics and reports **1546**

App Firewall logs **1549**

Appendices **1559**

PCRE character encoding format **1559**

Whitehat WASC signature types for WAF use **1562**

Streaming support for request processing **1563**

Trace HTML requests with security logs **1566**

App Firewall support for cluster configurations **1569**

Debugging and troubleshooting **1571**

High CPU **1571**

Memory **1573**

Large file upload failure **1575**

Learning **1575**

Signatures **1577**

Trace Log **1578**

Miscellaneous **1579**

References **1580**

Cache Redirection **1581**

Cache redirection policies **1581**

Built-in cache redirection policies **1582**

Configure a cache redirection policy **1585**

Cache redirection configurations **1593**

Configure transparent redirection **1593**

Enable cache redirection and load balancing **1594**

Configure edge mode **1595**

Configure a cache redirection virtual server **1596**

Bind policies to the cache redirection virtual server **1598**

Unbind a policy from a cache redirection virtual server **1599**

Create a load balancing virtual server **1600**

Configure an HTTP service **1602**

Bind/unbind a service to/from a load balancing virtual server **1604**

Disable the use the proxy port setting for transparent caching **1605**

Assign a port range to the NetScaler appliance **1605**

Enable load balancing virtual servers to redirect requests to cache **1606**

Configure forward proxy redirection **1608**

Create a DNS service **1609**

Create a DNS load balancing virtual server **1610**

Bind the DNS service to the virtual server **1611**

Configure a client web browser to use a forward proxy **1613**

Configure reverse proxy redirection **1613**

Selective cache redirection **1617**

Enable content switching **1618**

Configure a load balancing virtual server for the cache **1619**

Configure policies for content switching **1620**

Configure precedence for policy evaluation **1624**

Administer a cache redirection virtual server **1625**

View cache redirection virtual server statistics **1625**

Enable or disable a cache redirection virtual server **1627**

Direct policy hits to the cache instead of the origin **1628**

Back up a cache redirection virtual server **1630**

Manage client connections for a virtual server **1631**

N-Tier cache redirection **1636**

Configure the upper-tier NetScaler appliances **1642**

Configure the lower-tier NetScaler appliances **1644**

Clustering **1645**

Supportability matrix for NetScaler cluster **1646**

Prerequisites for cluster nodes **1652**

Cluster overview **1654**

Synchronization across cluster nodes **1655**

Striped, partially striped, and spotted configurations **1657**

Communication in a cluster setup **1660**

Traffic distribution in a cluster setup **1663**

Cluster nodegroups **1665**

Cluster and node states **1666**

Routing in a cluster **1666**

Setting up a NetScaler cluster **1668**

Setting up inter-node communication **1668**

Creating a NetScaler cluster **1670**

Adding a node to the cluster **1674**

Viewing the details of a cluster **1677**

Distributing traffic across cluster nodes **1678**

Using Equal Cost Multiple Path (ECMP) **1678**

Use Case: ECMP with BGP routing **1682**

Using cluster link aggregation **1684**

Static cluster link aggregation **1687**

Dynamic cluster link aggregation **1688**

Link redundancy in a cluster with LACP **1690**

Managing the NetScaler cluster **1691**

Configuring linksets **1692**

Nodegroups for spotted and partially-striped configurations **1696**

Behavior of nodegroups **1697**

Configuring nodegroups for spotted and partially-striped configurations **1698**

Configuring redundancy for nodegroups **1700**

Disabling steering on the cluster backplane **1703**

Synchronizing cluster configurations **1704**

Synchronizing time across cluster nodes **1705**

Synchronizing cluster files **1705**

Viewing the statistics of a cluster **1707**

Discovering NetScaler appliances **1708**

Disabling a cluster node **1708**

Removing a cluster node **1709**

Removing a node from a cluster deployed using cluster link aggregation **1711**

Detecting jumbo probe on a cluster **1711**

Route monitoring for dynamic routes in cluster **1712**

Monitoring cluster setup using SNMP MIB with SNMP link **1713**

Monitoring command propagation failures in a cluster deployment **1714**

IPv6 ready logo support for clusters **1715**

Managing cluster heartbeat messages **1720**

Configuring owner node response status **1720**

Monitor static route (MSR) support for inactive nodes in a spotted cluster configuration **1721**

VRRP interface binding in a single node active cluster **1722**

Cluster setup and usage scenarios **1723**

Creating a two-node cluster **1723**

Migrating an HA setup to a cluster setup **1723**

Transitioning between a L2 and L3 Cluster **1727**

Setting up GSLB in a cluster **1728**

Using cache redirection in a cluster **1730**

Using L2 mode in a cluster setup **1730**

Using cluster LA channel with linksets **1731**

Backplane on LA channel **1732**

Common interfaces for client and server and dedicated interfaces for backplane **1734**

Common switch for client, server, and backplane **1737**

Common switch for client and server and dedicated switch for backplane **1739**

Different switch for every node **1742**

Sample cluster configurations **1743**

Monitoring services in a cluster using path monitoring **1747**

Upgrading or downgrading the NetScaler cluster **1750**

Operations supported on individual cluster nodes **1753**

FAQs **1754**

Troubleshooting the NetScaler cluster **1762**

Tracing the packets of a NetScaler cluster **1763**

Troubleshooting common issues **1767**

Content Switching **1770**

Configuring basic content switching **1773**

Enabling content switching **1774**

Creating content switching virtual servers **1775**

Configuring a load balancing setup for content switching **1775**

Configuring a content switching action **1776**

Configuring content switching policies **1779**

Configuring content switching policy labels **1781**

Binding policies to a content switching virtual server **1784**

Configuring policy based logging for content switching **1785**

Verifying the configuration **1787**

Customizing the basic content switching configuration **1792**

Content switching for diameter protocol **1797**

Protecting the content switching setup against failure **1799**

Managing a content switching setup **1805**

Managing client connections **1809**

Troubleshooting **1813**

DataStream **1815**

Configuring Database Users **1817**

Configuring a Database Profile **1819**

Configuring Load Balancing for DataStream **1820**

Configuring Content Switching for DataStream **1822**

Configuring Monitors for DataStream **1823**

Use Case 1: Configuring DataStream for a Master/Slave Database Architecture **1825**

Use Case 2: Configuring the Token Method of Load Balancing for DataStream **1828**

Use Case 3: Logging MSSQL Transactions in Transparent Mode **1830**

Use Case 4: Database Specific Load Balancing **1833**

DataStream Reference **1845**

Domain Name System **1848**

Configure DNS resource records **1851**

Create SRV records for a service **1853**

Creating AAAA Records for a Domain Name **1854**

Creating address records for a Domain Name **1855**

Create MX records for a mail exchange server **1856**

Create NS Records for an authoritative server **1857**

Create CNAME records for a subdomain **1858**

Create NAPTR records for telecommunications domain **1859**

Create PTR records for IPv4 and IPv6 addresses **1860**

Create SOA records for authoritative information **1861**

Create TXT records for holding descriptive text **1862**

Viewing DNS statistics **1863**

Configure a DNS zone **1865**

Configure the NetScaler as an authoritative domain name server **1866**

Configure the NetScaler appliance as a DNS proxy server **1871**

Configure the NetScaler as an end resolver **1876**

Configure the NetScaler appliance as a forwarder **1879**

Add a name server **1880**

Set DNS lookup priority **1882**

Disable and enable name servers **1883**

Configure DNS logging **1883**

Configuring DNS suffixes **1897**

DNS ANY query **1898**

Configure negative caching of DNS records **1899**

Domain name system security extensions **1901**

Configure DNSSEC **1902**

Configure DNSSEC when the NetScaler is authoritative for a zone **1910**

Configure DNSSEC for a zone for which the NetScaler is a DNS proxy server **1911**

Configure DNSSEC for GSLB domain names **1913**

Zone Maintenance **1913**

Offload DNSSEC operations to the NetScaler **1917**

Firewall Load Balancing **1918**

Sandwich Environment **1920**

Enterprise Environment **1937**

Multiple-Firewall Environment **1949**

Global Server Load Balancing **1960**

How GSLB Works **1961**

GSLB Deployment Types **1962**

Active-Active Site Deployment **1962**

Active-Passive Site Deployment **1964**

Parent-Child Topology Deployment using the MEP Protocol **1965**

GSLB Configuration Entities **1971**

Configuring Global Server Load Balancing (GSLB) **1974**

Configuring an Authoritative DNS Service **1975**

Configuring a Basic GSLB Site **1976**

Configuring a GSLB Service **1978**

Configuring a GSLB Virtual Server **1980**

Binding GSLB Services to a GSLB Virtual Server **1984**

Binding a Domain to a GSLB Virtual Server **1984**

Synchronizing a Configuration in a GSLB Setup **1988**

Testing the GSLB setup **1993**

Configuring the Metrics Exchange Protocol (MEP) **1994**

Configuring Site-to-Site Communication **1998**

Customize your GSLB configuration **2002**

Changing the GSLB methods **2007**

Configuring Static Proximity **2008**

Add a location file to create a static proximity database **2009**

Add custom entries to a static proximity database **2012**

Set location qualifiers **2014**

Specify proximity method **2015**

Synchronize GSLB static proximity database **2016**

Configuring the Dynamic Method (RTT) **2017**

How to configure persistence in GSLB **2019**

Override static proximity behavior by configuring preferred locations **2024**

Monitoring GSLB Services **2026**

Monitoring GSLB Sites **2030**

Protecting the GSLB Setup Against Failure **2030**

Managing Client Connections **2036**

Configuring GSLB for Disaster Recovery **2045**

Configuring GSLB for Proximity **2051**

Example of a Complete Parent-Child Configuration Using the Metrics Exchange Protocol **2052**

Configuring GSLB Service Selection Using Content Switching **2058**

Configuring Global Server Load Balancing for DNS Queries with NAPTR records **2060**

Using the EDNS0 Client Subnet Option for Global Server Load Balancing **2063**

Link load balancing **2068**

Configuring a Basic LLB Setup **2069**

Configuring RNAT with LLB **2079**

Configuring a Backup Route **2081**

Resilient LLB Deployment Scenario **2084**

Monitoring an LLB Setup **2086**

Load Balancing **2088**

How load balancing works **2089**

Set up basic load balancing **2099**

Load balance virtual server and service states **2112**

Support for load balancing profile **2115**

Load balancing algorithms **2118**

Least connection method **2121**

Round robin method **2126**

Least response time method **2128**

LRTM method **2134**

Hashing methods **2140**

Least bandwidth method **2147**

Least packets method **2151**

Custom load method **2155**

Static proximity method **2160**

Token method **2161**

Configure a load balancing method that does not include a policy **2164**

Persistence and persistent connections **2165**

About Persistence **2165**

Source IP address persistence **2168**

HTTP cookie persistence **2168**

SSL session ID persistence **2171**

Diameter AVP number persistence **2171**

Custom server ID persistence **2172**

IP address persistence **2174**

SIP Call ID persistence **2175**

RTSP session ID persistence **2175**

Configure URL passive persistence **2176**

Configure persistence based on user-defined rules **2177**

Configure persistence types that do not require a rule **2179**

Configure backup persistence **2180**

Configure persistence groups **2181**

Configure RADIUS load balancing with persistence **2183**

View persistence sessions **2186**

Clear persistence sessions **2187**

Override persistence settings for overloaded services **2188**

Troubleshooting **2190**

Insert cookie attributes to ADC generated cookies **2192**

Customize a load balancing configuration **2203**

Customize the hash algorithm for persistence across virtual servers **2204**

Configure the redirection mode **2207**

Configure per-VLAN wildcarded virtual servers **2208**

Assign weights to services **2209**

Configure the MySQL and Microsoft SQL server version setting **2211**

Configure diameter load balancing **2212**

Configure FIX load balancing **2218**

Protect a load balancing configuration against failure **2224**

Redirect client requests to an alternate URL **2224**

Configure a backup load balancing virtual server **2225**

Configure spillover **2226**

Connection failover **2234**

Flush the surge queue **2238**

Manage a load balancing setup **2240**

Manage server objects **2241**

Manage services **2242**

Manage a load balancing virtual server **2243**

Load balancing visualizer **2244**

Manage client traffic **2247**

Configure sessionless load balancing virtual servers **2248**

Redirect HTTP requests to a cache **2250**

Direct requests according to priority **2251**

Direct requests to a custom web page **2252**

Enable cleanup of virtual server connections **2253**

Rewrite ports and protocols for HTTP redirection **2255**

Insert IP address and port of a virtual server in the request header **2260**

Use a specified source IP for backend communication **2261**

Set a time-out value for idle client connections **2268**

Manage RTSP connections **2268**

Manage client traffic on the basis of traffic rate **2269**

Identify a connection with layer 2 parameters **2270**

Configure the prefer direct route option **2271**

Use a source port from a specified port range for backend communication **2271**

Configure source IP persistency for backend communication **2273**

Advanced load balancing settings **2274**

Gradually stepping up the load on a new service with virtual server–level slow start **2275**

The no-monitor option for services **2281**

Protect applications on protected servers against traffic surges **2284**

Enable cleanup of virtual server and service connections **2285**

Graceful shutdown of services **2287**

Direct requests to a custom web page **2291**

Enable access to services when down **2292**

Enable TCP buffering of responses **2293**

Enable compression **2293**

Maintain client connection for multiple client requests **2294**

Insert the IP address of the client in the request header **2295**

Use source IP address of the client when connecting to the server **2296**

Configure the source port for server-side connections **2296**

Set a limit on the number of client connections **2299**

Set a limit on number of requests per connection to the server **2300**

Set a threshold value for the monitors bound to a service **2300**

Set a timeout value for idle client connections **2301**

Set a timeout value for idle server connections **2302**

Set a limit on the bandwidth usage by clients **2303**

Redirect client requests to a cache **2303**

Retain the VLAN identifier for VLAN transparency **2304**

Configure automatic state transition based on percentage health of bound services **2305**

Built-in monitors **2306**

TCP-based application monitoring **2306**

SSL service monitoring **2308**

FTP service monitoring **2310**

Secure monitoring of servers by using SFTP **2311**

Set SSL parameters on a secure monitor **2312**

SIP service monitoring **2314**

RADIUS service monitoring **2314**

Monitor accounting information delivery from a RADIUS server **2315**

DNS and DNS-TCP service monitoring **2316**

LDAP service monitoring **2317**

MySQL service monitoring **2318**

SNMP service monitoring **2319**

NNTP service monitoring **2319**

POP3 service monitoring **2320**

SMTP service monitoring **2321**

RTSP service monitoring **2321**

XML broker service monitoring **2327**

ARP request monitoring **2327**

XenDesktop Delivery Controller service monitoring **2328**

Citrix StoreFront stores monitoring **2330**

Custom monitors **2331**

Configure HTTP-inline monitors **2331**

Understand user monitors **2332**

How to use a user monitor to check web sites **2338**

Understand the internal dispatcher **2339**

Configure a user monitor **2341**

Understand load monitors **2342**

Configure load monitors **2344**

Unbind metrics from a metrics table **2345**

Configure reverse monitoring for a service **2345**

Configure monitors in a load balancing setup **2348**

Create monitors **2349**

Bind monitors to services **2349**

Modify monitors **2350**

Enable and disable monitors **2351**

Unbind monitors **2352**

Remove monitors **2353**

View monitors **2353**

Close monitor connections **2354**

Ignore the upper limit on client connections for monitor probes **2356**

Manage a large scale deployment **2357**

Ranges of virtual servers and services **2357**

Configure service groups **2360**

Manage service groups **2363**

Configure automatic domain based service group scaling **2370**

Translate the IP address of a domain-based server **2373**

Mask a virtual server IP address **2374**

Configure load balancing for commonly used protocols **2376**

Load balance a group of FTP servers **2377**

Load balance DNS servers **2379**

Load balance domain-name based services **2382**

Load balance a group of SIP servers **2385**

Load balance RTSP servers **2395**

Load balance remote desktop protocol (RDP) servers **2398**

Use case 1: SMPP load balancing **2402**

Use case 2: Configure rule based persistence based on a name-value pair in a TCP byte stream **2412**

Use case 3: Configure load balancing in direct server return mode **2414**

Use case 4: Configure LINUX servers in DSR mode **2418**

Use case 5: Configure DSR mode when using TOS **2418**

Use case 6: Configure load balancing in DSR mode for IPv6 networks by using the TOS field **2424**

Use case 7: Configure load balancing in DSR mode by using IP Over IP **2427**

Use case 8: Configure load balancing in one-arm mode **2431**

Use case 9: Configure load balancing in the inline mode **2433**

Use case 10: Load balancing of intrusion detection system servers **2433**

Use case 11: Isolating network traffic using listen policies **2438**

Use case 12: Configure XenDesktop for load balancing **2444**

Use case 13: Configure XenApp for load balancing **2446**

Use case 14: ShareFile wizard for load balancing Citrix ShareFile **2449**

Troubleshooting **2453**

Load balancing FAQs **2459**

Networking **2461**

IP Addressing **2462**

Configuring NetScaler-Owned IP Addresses **2462**

Configuring the NetScaler IP Address (NSIP) **2463**

Configuring and Managing Virtual IP (VIP) Addresses **2465**

Configuring ARP response Suppression for Virtual IP addresses (VIPs) **2470**

Configuring Subnet IP Addresses (SNIPs) **2473**

Configuring GSLB Site IP Addresses (GSLBIP) **2479**

Removing a NetScaler-Owned IP Address **2479**

Configuring Application Access Controls **2480**

How the NetScaler Proxies Connections **2482**

Enable Use Source IP Mode **2484**

Configuring Network Address Translation **2487**

INAT **2487**

Coexistence of INAT and Virtual Servers **2489**

Stateless NAT46 **2490**

DNS64 **2495**

Stateful NAT64 Translation **2500**

RNAT **2504**

Configuring Prefix-Based IPv6-IPv4 Translation **2515**

IP Prefix NAT **2516**

Static ARP **2518**

Set the Timeout for Dynamic ARP Entries **2520**

Neighbor Discovery **2521**

IP Tunnels **2523**

Interfaces **2530**

Configuring MAC-Based Forwarding **2530**

Configuring Network Interfaces **2532**

Configuring Forwarding Session Rules **2535**

Understanding VLANs **2540**

Configuring a VLAN **2542**

Configuring VLANs on a Single Subnet **2545**

Configuring VLANs on Multiple Subnets **2546**

Configuring Multiple Untagged VLANs across Multiple Subnets **2547**

Configuring Multiple VLANs with 802.1q Tagging **2548**

Configuring NSVLAN **2549**

Configuring Allowed VLAN List **2552**

Configuring Bridge Groups **2553**

Configuring Virtual MACs **2555**

Configuring Link Aggregation **2555**

Redundant Interface Set **2563**

Binding an SNIP address to an Interface **2568**

Monitor the Bridge Table and Changing the Aging time **2573**

NetScaler Appliances in Active-Active Mode Using VRRP **2573**

Configuring Active-Active Mode **2577**

Configuring Send to Master **2580**

Configuring VRRP Communication Intervals **2583**

Configuring Health Tracking based on Interface State **2590**

Delaying Preemption **2593**

Keeping a VIP Address in the Backup State **2597**

Network Visualizer **2597**

Configuring Link Layer Discovery Protocol **2598**

Jumbo Frames **2602**

Configuring Jumbo Frames Support on a NetScaler Appliance **2602**

Use Case 1 – Jumbo to Jumbo Setup **2604**

Use Case 2 – Non-Jumbo to Jumbo Setup **2608**

Use Case 3 – Coexistence of Jumbo and Non-Jumbo flows on the Same Set of Interfaces **2612**

NetScaler Support for Microsoft Direct Access Deployment **2615**

Access Control Lists **2618**

Simple ACLs and Simple ACL6s **2620**

Extended ACLs and Extended ACL6s **2625**

MAC Address Wildcard Mask for ACLs **2635**

Blocking Traffic on Internal Ports **2636**

IP Routing **2637**

Configuring Dynamic Routes **2637**

Configuring RIP **2640**

Configuring OSPF **2643**

Configuring BGP **2647**

Configuring IPv6 RIP **2658**

Configuring IPv6 OSPF **2660**

Configuring ISIS **2665**

Install Routes to the NetScaler Routing Table **2670**

Advertisement of SNIP and VIP Routes to Selective Areas **2671**

Configuring Static Routes **2673**

Route Health Injection Based on Virtual Server Settings **2678**

Configuring Policy-Based Routes **2680**

Policy-Based Routes (PBR) for IPv4 Traffic **2681**

Policy-Based Routes (PBR6) for IPv6 Traffic **2688**

MAC Address Wildcard Mask for PBRs **2690**

Using NULL Policy Based Routes to Drop Outgoing Packets **2692**

Troubleshooting Routing Issues **2693**

Generic Routing FAQs **2693**

Troubleshooting OSPF-Specific Issues **2695**

Internet Protocol version 6 (IPv6) **2696**

Traffic Domains **2703**

Inter Traffic Domain Entity Bindings **2711**

virtual MAC Based Traffic Domains **2711**

VXLAN **2716**

Optimization **2727**

Client Keep-Alive **2728**

HTTP compression **2730**

Integrated caching **2739**

Front end optimization **2747**

Content accelerator **2753**

Media classification **2757**

Reputation **2761**

IP Reputation **2762**

SSL offload and acceleration **2770**

SSL offloading configuration **2771**

How-to articles **2804**

SSL certificates **2804**

Create a certificate **2805**

Install, link, and update certificates **2810**

Generate a server test certificate **2835**

Import and convert SSL files **2836**

SSL profiles **2841**

SSL profile infrastructure **2842**

Appendix A: Sample migration of the SSL configuration after upgrade **2856**

Appendix B: Default front-end and back-end SSL profile settings **2856**

Legacy SSL profile **2860**

Certificate revocation lists **2863**

Monitor certificate status with OCSP **2870**

OCSP stapling **2875**

Ciphers available on the NetScaler appliances **2881**

ECDHE ciphers **2881**

Diffie-Hellman (DH) parameters generation and achieving PFS with DHE **2883**

Cipher redirection **2885**

Leverage hardware and software to improve ECDHE cipher performance **2887**

ECDSA cipher suites support **2889**

Configure user-defined cipher groups on the ADC appliance **2893**

Server certificate support matrix on the ADC appliance **2899**

Client authentication **2900**

Server authentication **2906**

SSL actions and policies **2910**

SSL policies **2911**

SSL built-in actions and user-defined actions **2912**

SSL policy binding **2915**

SSL policy labels **2918**

Support for DTLS protocol **2919**

Support for Intel Coleto SSL chip based platforms **2931**

MPX 9700/10500/12500/15500 FIPS appliances 2932

Configure FIPS appliances in a high availability setup 2942

Update the firmware to version 2.2 on a FIPS card 2945

Reset a locked HSM 2949

MPX 14000 FIPS appliances 2950

SDX 14000 FIPS appliances 2957

Limitations 2958

Terminology 2958

Initialize the HSM 2958

Create partitions 2960

Provision a new instance or modify an existing instance and assign a partition 2962

Configure the HSM for an instance on an SDX 14030/14060/14080 FIPS appliance 2963

Create a FIPS key for an instance on an SDX 14030/14060/14080 FIPS appliance 2966

Upgrade the FIPS firmware 2969

Support for Thales nShield® HSM 2971

Architecture overview 2972

Prerequisites 2974

Configure the ADC-Thales integration 2975

Limitations 2992

Appendix 2993

Support for Gemalto SafeNet Network hardware security module 2995

Prerequisites 2996

Configure a SafeNet client on the ADC 2996

Configure Safenet HSMs in a high availability setup on the ADC 3000

Additional ADC configuration **3004**

NetScaler appliances in a high availability setup **3005**

Limitations **3006**

Appendix **3007**

FAQ **3010**

Troubleshooting **3010**

SSL FAQs **3011**

Security **3030**

Content filtering **3031**

Enabling content filtering **3032**

Configure a content filtering action **3033**

Configure a content filtering policy **3035**

Binding a content filtering policy **3039**

Configuring content filtering for a commonly used deployment scenario **3041**

Troubleshooting **3044**

HTTP denial-of-service protection **3045**

Layer 3-4 SYN Denial-of-Service protection **3046**

Enable HTTP DoS protection **3048**

Define an HTTP DoS policy **3049**

Configuring an HTTP DoS service **3050**

Bind an HTTP DoS monitor and policy **3052**

Tuning the client detection/JavaScript challenger response rate **3055**

Guidelines for HTTP DoS protection deployment **3056**

Priority queuing **3056**

Enable priority queuing **3058**

Configure a priority queuing policy **3059**

Bind a priority queuing policy **3061**

Set up weighted queuing **3062**

SureConnect **3063**

Installing SureConnect **3064**

Configure SureConnect **3066**

Activating SureConnect **3074**

SureConnect environments **3075**

Surge protection **3084**

Disable and reenable surge protection **3086**

Set thresholds for surge protection **3088**

Flush the surge queue **3091**

DNS security options **3093**

System **3097**

Basic operations **3098**

Clock synchronization **3100**

Session timeout **3102**

System date and time **3104**

Backup and restore **3105**

Restart or shut down **3110**

Generate technical support bundle **3111**

Authentication and authorization **3115**

Configuring users, user groups, and command policies **3116**

Resetting the default administrator password **3123**

Example of a user scenario **3125**

TCP Configurations **3127**

HTTP configurations **3139**

Configuring HTTP/2 on the NetScaler appliance **3143**

SNMP **3147**

Configuring the NetScaler to generate SNMP traps **3149**

Configuring NetScaler for SNMP v1 and v2 queries **3154**

Configuring NetScaler for SNMPv3 queries **3157**

Configuring SNMP Alarms for rate limiting **3161**

Audit logging **3163**

Configuring NetScaler appliance for audit logging **3165**

Installing and configuring the NSLOG server **3170**

Running the NSLOG server **3176**

Customizing logging on the NSLOG server **3176**

SYSLOG Over TCP **3180**

Load balancing SYSLOG servers **3184**

Default settings for the log properties **3186**

Sample configuration file (audit.conf) **3187**

Web server logging **3188**

Configuring the NetScaler for web server logging **3188**

Installing the NetScaler web logging (NSWL) client **3190**

Configure NSWL client **3196**

Customize logging on the NSWL client system **3199**

Call Home **3213**

Reporting tool **3222**

CloudBridge Connector **3232**

Monitoring CloudBridge Connector tunnels **3235**

Configuring a CloudBridge Connector tunnel between two datacenters **3237**

Configuring CloudBridge Connector between datacenter and AWS cloud **3244**

Configuring a CloudBridge Connector tunnel between a datacenter and Azure cloud **3252**

Configuring CloudBridge Connector tunnel between datacenter and softlayer enterprise cloud **3264**

Configuring a CloudBridge Connector tunnel between a NetScaler appliance and Cisco IOS device **3265**

Configuring a CloudBridge Connector tunnel between a NetScaler appliance and fortinet fortiGate appliance **3275**

CloudBridge Connector tunnel diagnostics and troubleshooting **3283**

High Availability **3286**

Points to consider for a high availability setup **3287**

Configuring high availability **3289**

Configuring the communication intervals **3292**

Configuring synchronization **3293**

Synchronizing configuration files in a high availability setup **3294**

Configuring command propagation **3295**

Restricting high availability synchronization traffic to a VLAN **3296**

Configuring fail-safe mode **3297**

Configuring Virtual MAC Addresses **3299**

Configuring high availability nodes in different subnets **3303**

Configuring route monitors 3307

Limiting failovers caused by route monitors in non-INC mode 3310

Configuring failover interface set 3312

Understanding the causes of failover 3314

Forcing a node to fail over 3315

Forcing the secondary node to stay secondary 3316

Forcing the primary node to stay primary 3317

Understanding the high availability health check computation 3318

High availability FAQ 3319

Troubleshooting high availability issues 3321

Managing high availability heartbeat messages on a NetScaler appliance 3324

TCP optimization 3325

Reference Material 3337

NetScaler Release Notes

July 19, 2021

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

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

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

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

To view the release notes document for a specific build of release 11.1, click the corresponding link in the following table. When the release notes are updated for a build, the version number of the release notes and the publish date are also updated. The release notes publish date might not be the same as the build GA date.

Release notes for NetScaler release 11.1	Release notes publish date	Release notes version
Build 65.22	July 19, 2021	1.0
Build 65.12	September 17, 2020	1.0
Build 65.10	July 16, 2020	1.0
Build 64.14	July 07, 2020	1.0
Build 64.11	March 23, 2020	1.0
Build 63.15	Jan 22, 2020	2.0
Build 63.9	January 22, 2020	3.0
Build 62.8	July 29, 2019	2.0
Build 61.7	Feb 1, 2019	3.0
Build 60.14	October 30, 2018	1.0
Build 59.10	September 5, 2018	4.0

Release notes for NetScaler release 11.1	Release notes publish date	Release notes version
Build 58.13	June 13, 2018	7.0

Getting started with Citrix ADC

September 21, 2020

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

Understanding Citrix ADC

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

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

Data switching

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

Firewall security

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

Optimization

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

Policy infrastructure

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

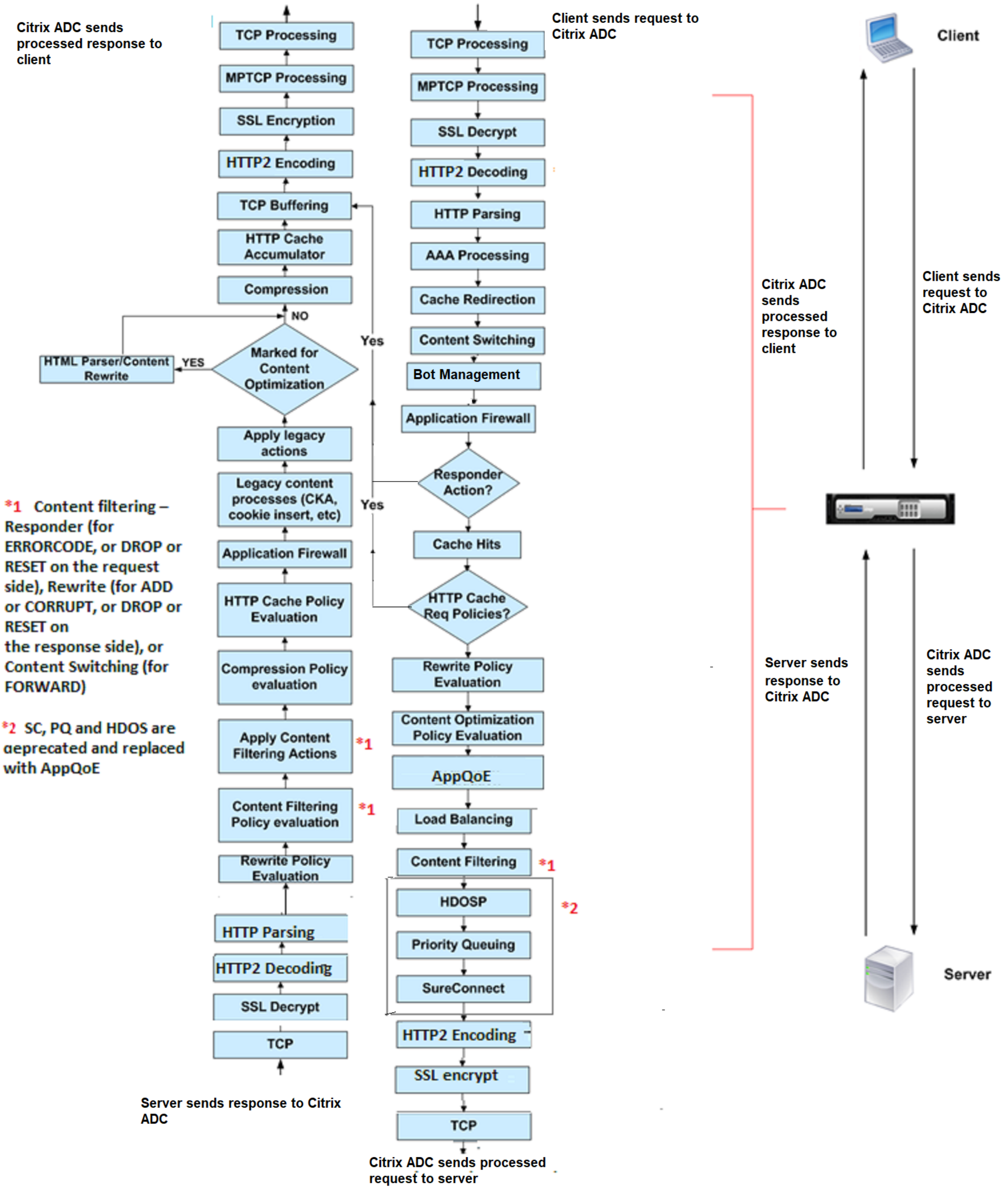
When an ADC appliance receives traffic, the appropriate policy list determines how to process the traffic. Each policy on the list contains one or more expressions, which together define the criteria that a connection must meet to match the policy.

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

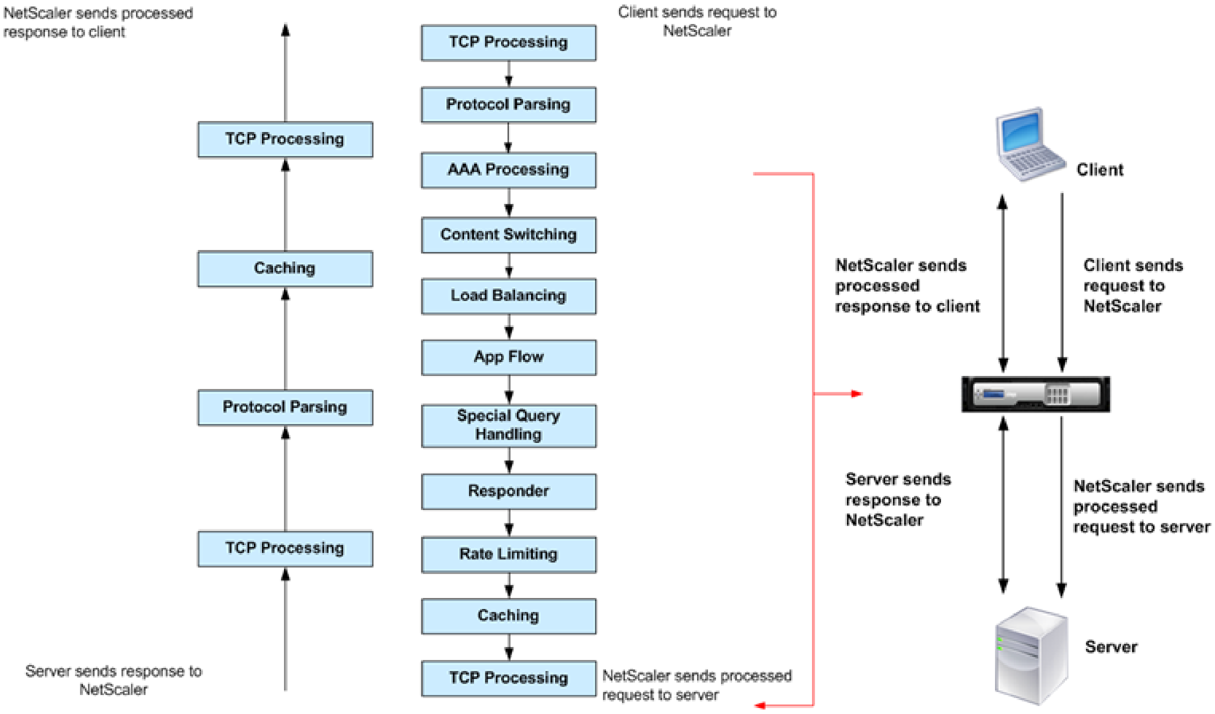
Packet flow

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

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


Citrix ADC sends processed response to client
TCP Processing
MPTCP Processing
SSL Encryption
HTTP2 Encoding
TCP Buffering
HTTP Cache Accumulator
Compression
NO
HTML Parser/Content Rewrite
YES
Marked for Content Optimization
Apply legacy actions
Legacy content processes (CKA, cookie insert, etc)
Application Firewall
HTTP Cache Policy Evaluation
Compression Policy evaluation
Apply Content Filtering Actions
*1
Content Filtering Policy evaluation
*1
Rewrite Policy Evaluation
HTTP Parsing
HTTP2 Decoding
SSL Decrypt
TCP
Server sends response to Citrix ADC
*1 Content filtering – Responder (for ERRORCODE, or DROP or RESET on the request side), Rewrite (for ADD or CORRUPT, or DROP or RESET on the response side), or Content Switching (for FORWARD)
*2 SC, PQ and HDOS are deprecated and replaced with AppQoE
Yes
Yes
Client sends request to Citrix ADC
TCP Processing
MPTCP Processing
SSL Decrypt
HTTP2 Decoding
HTTP Parsing
AAA Processing
Cache Redirection
Content Switching
Bot Management
Application Firewall
Responder Action?
Cache Hits
HTTP Cache Req Policies?
Rewrite Policy Evaluation
Content Optimization Policy Evaluation
AppQoE
Load Balancing
Content Filtering
*1
HDOSP
*2
Priority Queuing
SureConnect
HTTP2 Encoding
SSL encrypt
TCP
Citrix ADC sends processed request to server
Citrix ADC sends processed response to client
Client
Client sends request to Citrix ADC
Server sends response to Citrix ADC
Citrix ADC sends processed request to server
Server

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


NetScaler sends processed response to client
TCP Processing
Caching
Protocol Parsing
TCP Processing
Server sends response to NetScaler
TCP Processing
Protocol Parsing
AAA Processing
Content Switching
Load Balancing
App Flow
Special Query Handling
Responder
Rate Limiting
Caching
TCP Processing
Client sends request to NetScaler
NetScaler sends processed request to server
Client
NetScaler sends processed response to client
Client sends request to NetScaler
Server sends response to NetScaler
NetScaler sends processed request to server
Server

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

System limitation

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

Where does a NetScaler appliance fit in the network?

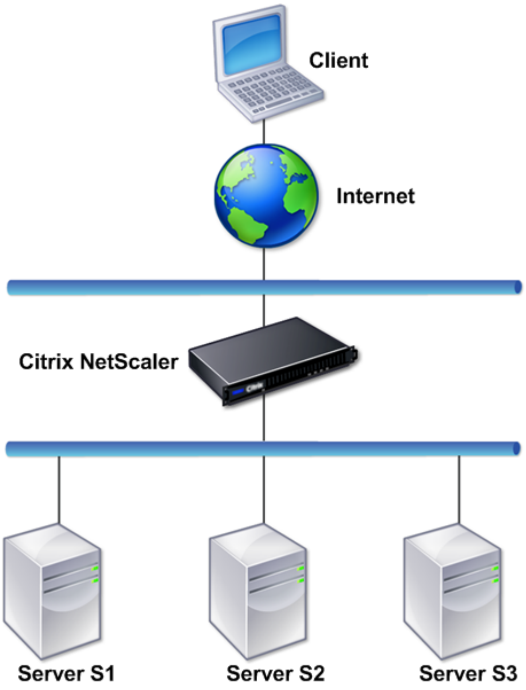
September 21, 2020

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

Physical deployment modes

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

Figure 1. Inline Deployment



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

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

NetScaler as an L2 device

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

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

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

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

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

NetScaler as a packet forwarding device

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

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

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

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

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

How a NetScaler appliance communicates with clients and servers

September 21, 2020

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

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

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

Understanding NetScaler-owned IP addresses

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

- NetScaler IP (NSIP) address

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

- Virtual server IP (VIP) address

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

- Subnet IP (SNIP) address

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

- IP Set

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

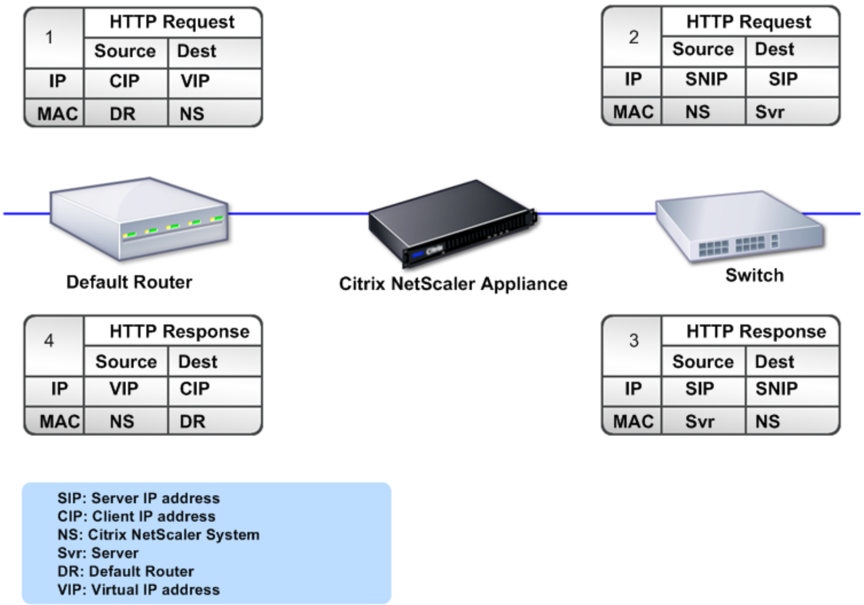
- Net Profile

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

How Traffic flows are managed

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

Figure 1. Virtual Server Based Connections


1
HTTP Request
Source Dest
IP CIP VIP
MAC DR NS
2
HTTP Request
Source Dest
IP SNIP SIP
MAC NS Svr
Default Router
Citrix NetScaler Appliance
Switch
4
HTTP Response
Source Dest
IP VIP CIP
MAC NS DR
3
HTTP Response
Source Dest
IP SIP SNIP
MAC Svr NS
SIP: Server IP address
CIP: Client IP address
NS: Citrix NetScaler System
Svr: Server
DR: Default Router
VIP: Virtual IP address

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

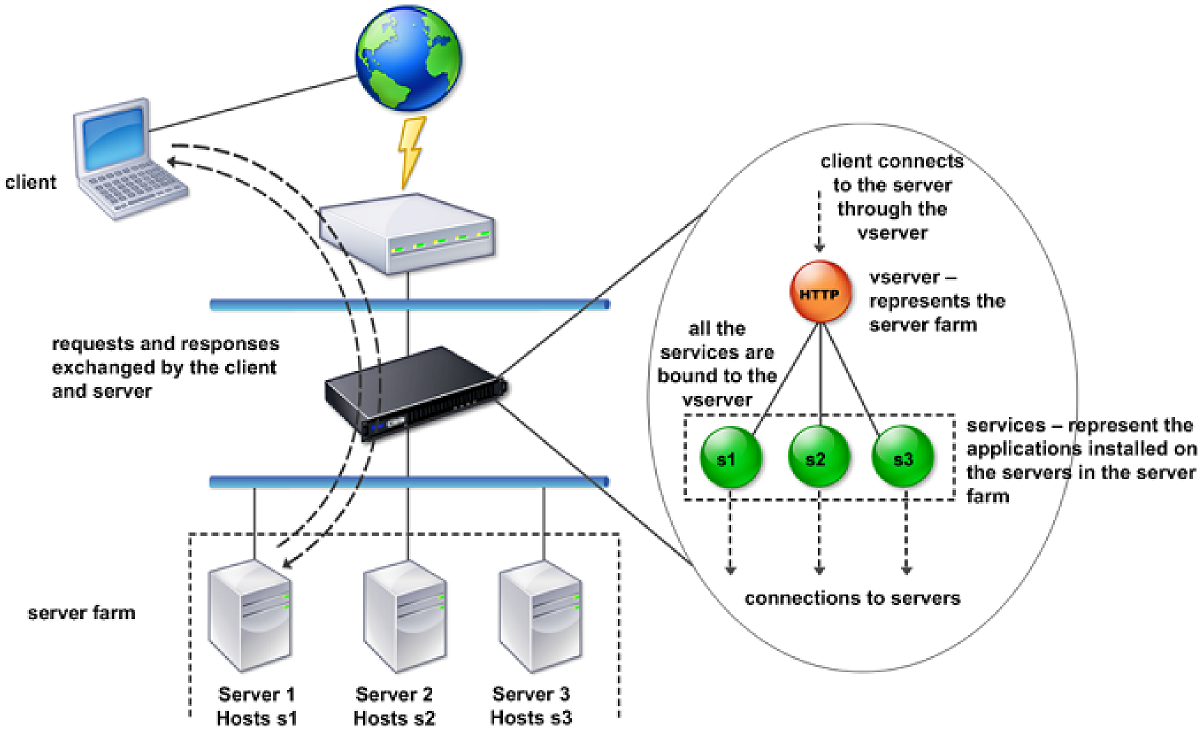
For cases in which the servers need the actual client IP address, the appliance can be configured to modify the HTTP header by inserting the client IP address as an additional field, or configured to use the client IP address instead of a SNIP address for connections to the servers.

Traffic management building blocks

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

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

Figure 2. How traffic management building blocks work

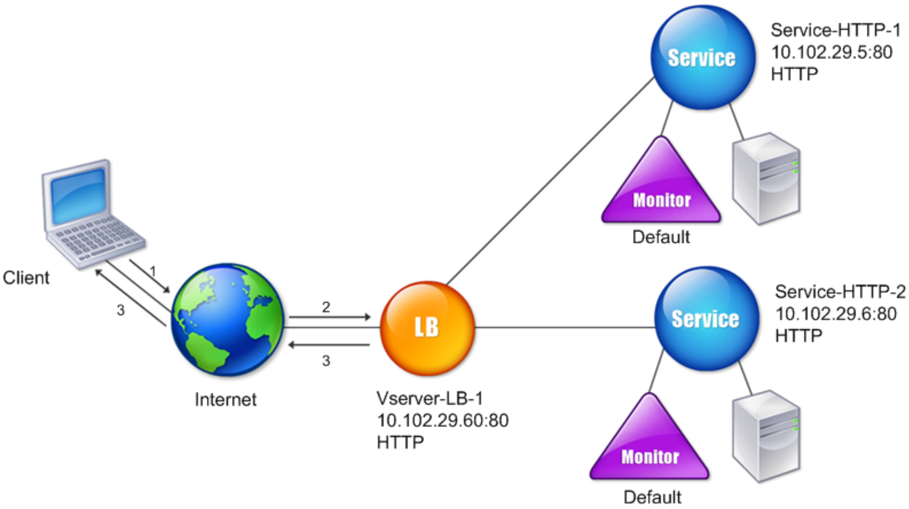

client
requests and responses exchanged by the client and server
server farm
Server 1 Hosts s1
Server 2 Hosts s2
Server 3 Hosts s3
client connects to the server through the vserver
HTTP
vserver – represents the server farm
all the services are bound to the vserver
s1
s2
s3
services – represent the applications installed on the servers in the server farm
connections to servers

A simple load balancing configuration

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

The basic building blocks of a typical load balancing configuration are services and load balancing virtual servers. The services represent the applications on the servers. The virtual servers abstract the servers by providing a single IP address to which the clients connect. To ensure that client requests are sent to a server, you need to bind each service to a virtual server. That is, you must create services for every server and bind the services to a virtual server. Clients use the VIP address to connect to a NetScaler appliance. When the appliance receives client requests sent to the VIP address, it sends them to a server determined by the load balancing algorithm. Load balancing uses a virtual entity called a monitor to track whether a specific configured service (server plus application) is available to receive requests.

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


Service
Service-HTTP-1
10.102.29.5:80
HTTP
Monitor
Default
Client
1
3
2
3
LB
Internet
Vserver-LB-1
10.102.29.60:80
HTTP
Service
Service-HTTP-2
10.102.29.6:80
HTTP
Monitor
Default

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

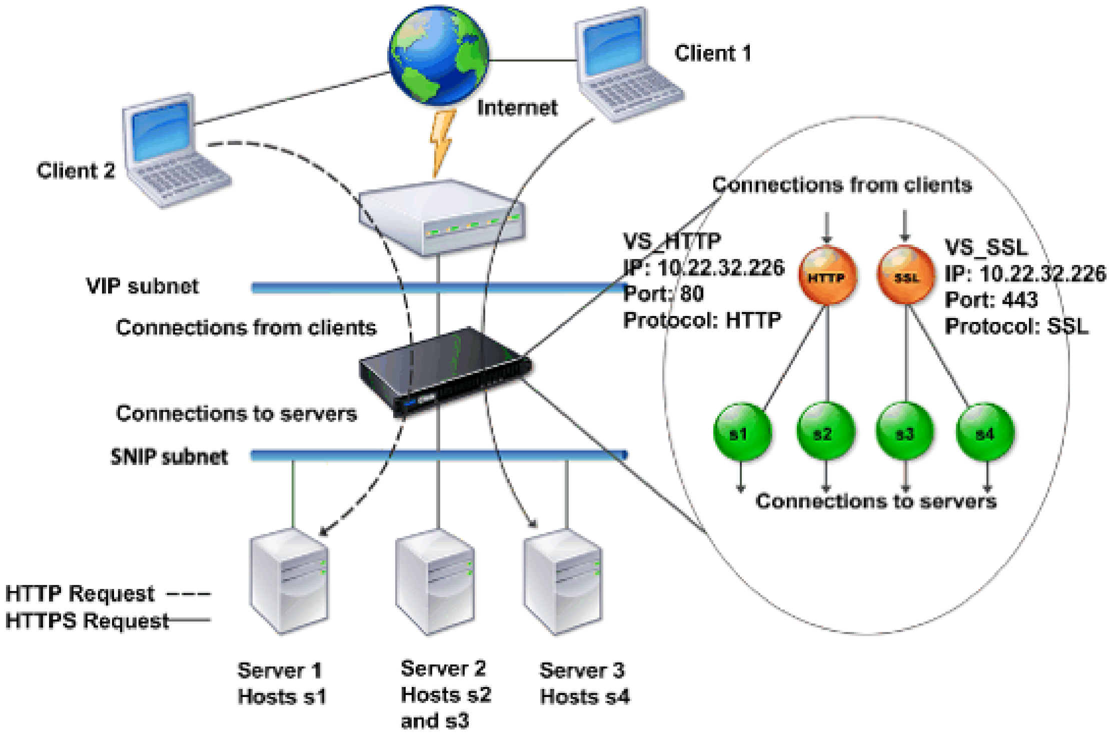
Understanding virtual servers

A virtual server is a named NetScaler entity that external clients can use to access applications hosted on the servers. It is represented by an alphanumeric name, virtual IP (VIP) address, port, and protocol. The name of the virtual server is of only local significance and is designed to make the virtual server easier to identify. When a client attempts to access applications on a server, it sends a request to the VIP instead of the IP address of the physical server. When the appliance receives a request at the VIP address, it terminates the connection at the virtual server and uses its own connection with the server on behalf of the client. The port and protocol settings of the virtual server determine the applications that the virtual server represents. For example, a web server can be represented by a virtual server and a service whose port and protocol are set to 80 and HTTP, respectively. Multiple virtual servers can use the same VIP address but different protocols and ports.

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

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

Figure 4. Multiple Virtual Servers with a Single VIP Address


Client 1
Internet
Client 2
Connections from clients
VS_HTTP
IP: 10.22.32.226
Port: 80
Protocol: HTTP
HTTP
SSL
VS_SSL
IP: 10.22.32.226
Port: 443
Protocol: SSL
VIP subnet
Connections from clients
s1
s2
s3
s4
Connections to servers
SNIP subnet
Connections to servers
HTTP Request
HTTPS Request
Server 1
Hosts s1
Server 2
Hosts s2
and s3
Server 3
Hosts s4

The preceding figure shows a configuration consisting of two virtual servers with a common VIP address but different ports and protocols. Each of the virtual servers has two services bound to it. The services s1 and s2 are bound to VS_HTTP and represent the HTTP applications on Server 1 and Server 2. The services s3 and s4 are bound to VS_SSL and represent the SSL applications on Server 2 and Server 3 (Server 2 provides both HTTP and SSL applications). When the appliance receives an HTTP request at the VIP address, it processes the request as specified by the settings of VS_HTTP and sends it to either Server 1 or Server 2. Similarly, when the appliance receives an HTTPS request at the VIP address, it processes it as specified by the settings of VS_SSL and it sends it to either Server 2 or Server 3.

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

Virtual servers can be grouped into the following categories:

- Load balancing virtual server

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

- Cache redirection virtual server

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

- Content switching virtual server

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

- Virtual private network (VPN) virtual server

  Decrypts tunneled traffic and sends it to intranet applications.

- SSL virtual server

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

Understanding services

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

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

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

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

Introduction to the NetScaler product line

September 21, 2020

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

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

NetScaler Hardware Platforms

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

NetScaler MPX hardware platform

NetScaler SDX hardware platform

NetScaler Editions

The NetScaler operating system is available in three editions:

- Standard
- Enterprise
- Platinum

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

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

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

Supported releases on NetScaler hardware

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

NetScaler MPX Hardware-Software Compatibility Matrix

NetScaler SDX Hardware-Software Compatibility Matrix

Supported browsers

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

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

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

Install the hardware

September 21, 2020

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

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

Task

Description

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

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

2. Prepare for installation

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

3. Install the hardware

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

4. Configure the appliance.

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

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

- NetScaler MPX hardware documentation
- NetScaler SDX hardware documentation

Access a NetScaler appliance

September 23, 2020

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

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

- Did you select an NSIP that is an IP address already assigned to another device on your network?
- Did you assign the same NSIP to multiple NetScaler appliances?

- The NSIP is reachable on all physical ports. The ports on a NetScaler are host ports, not switch ports.

The following table summarizes the available access methods.

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

Command line interface

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

Log on to the Command Line Interface through the Console Port

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

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

1. Connect the console port to a serial port on the workstation. For more information, see Connect the console cable.
2. On the workstation, start HyperTerminal or any other terminal emulation program. If the logon prompt does not appear, you might have to press ENTER one or more times to display it.
3. In User Name, type `nsroot`. In Password, type `nsroot` and if that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance.

Log on to the Command Line Interface by using SSH

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

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

- PuTTY

  Open Source software supported on multiple platforms. Available at:

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

- `Vandyke` Software SecureCRT

  Commercial software supported on the Windows platform. Available at:

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

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

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

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

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

```
login as: nsroot


Using keyboard-interactive authentication.


Password:


Last login: Tue Jun 16 10:37:28 2009 from 10.102.29.9






Done


>

<!--NeedCopy-->
```

NetScaler GUI

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

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

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

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

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

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

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

The following browsers are supported.

Operating System:
Windows 7

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

Operating System:
Windows 64 bit

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

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

Use the NetScaler GUI

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

To log on to the GUI, follow these steps:

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

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

2. In the **User Name** text box, type `nsroot`.

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

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

Use the Statistical Utility

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

To log on to Dashboard, follow these steps:

1. Open your web browser and enter the NSIP as an HTTP address. The Citrix Logon page appears.
2. In the User Name text box, type `nsroot`.
3. In the **Password** text box, type the administrative password you assigned to the `nsroot` account during initial configuration. If that password does not work, try typing the serial number of the appliance. The serial number bar code is available at the back of the appliance.

Configure the ADC for the first time

September 21, 2020

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

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

NITRO API

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

Secure your NetScaler deployment

September 21, 2020

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

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

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

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

Configure high availability

September 21, 2020

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

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

Figure 1. High availability in one-arm mode

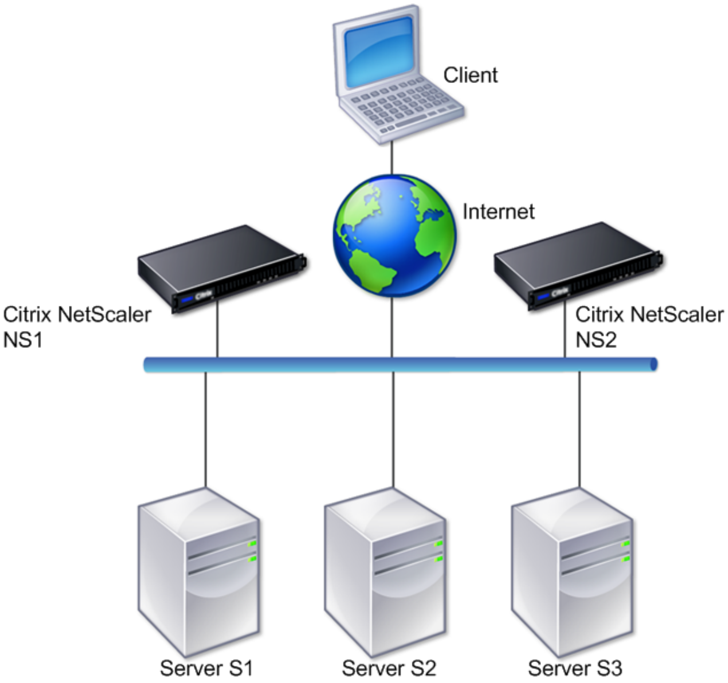
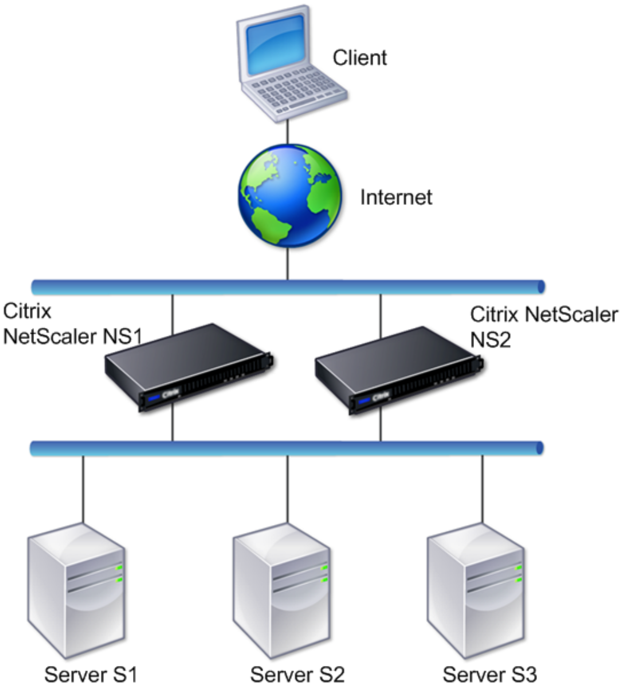

Client
Internet
Citrix NetScaler
NS1
Citrix NetScaler
NS2
Server S1
Server S2
Server S3

Figure 2. High availability in two-arm mode



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

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

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

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

Add a Node

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

Through CLI

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

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

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

  Example

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

Through GUI

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

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

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

Disable high availability monitoring for unused interfaces

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

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

Through CLI

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

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

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

  Example

```
> set interface 1/8 -haMonitor OFF
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4000 <ENABLED, DOWN, down, autoneg, 802.1q>
        MTU=1514, native vlan=1, MAC=00:d0:68:15:fd:3d, downtime 238h55m44s
        Requested: media AUTO, speed AUTO, duplex AUTO, fctl OFF,
                 throughput 0

        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
<!--NeedCopy-->
```

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

Through GUI

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

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

Change an RPC node password

October 14, 2020

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

To change an RPC node password by using the GUI

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

   > **Note**
   >
   > When you enable the **Secure** option, the appliance encrypts all communication sent from the node to other RPC nodes thus securing the RPC communication.

To change an RPC node password by using the CLI

At the command line, type the following commands:

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

Example:

```
> set ns rpcNode 192.0.2.4 -password mypassword -secure YES
 Done
> show rpcNode
.
.
.
 IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
     SrcIP:  *             Secure:  ON
Done
>

<!--NeedCopy-->
```

Configure a FIPS appliance for the first time

September 21, 2020

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

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

Configure secure HTTPS by using the CLI

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

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

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

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

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

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

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

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

Configure secure HTTPS by using the GUI

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

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

11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select nshttps-::11-443 from the list, and then click Open.
13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Click OK.

Configure secure RPC by using the CLI

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

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

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

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

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

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

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

   bind ssl service nskrpcs-127.0.0.1-3009 -certkeyname server

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

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

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

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

Configure secure RPC by using the GUI

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

1. Initialize the hardware security module (HSM) on the FIPS card of the appliance. For information about initializing the HSM, see Configure the HSM.

2. Enable the secure information system (SIM). For information about enabling the SIM on the primary and secondary appliances, Configure FIPS appliances in a high availability setup.
3. Import the FIPS key into the HSM of the FIPS card of the appliance. For more information about importing a FIPS key, the Import an existing FIPS key section.
4. Navigate to Traffic Management > SSL > Certificates.
5. In the details pane, click Install.
6. In the Install Certificate dialog box, type the certificate details.
7. Click Create, and then click Close.
8. Navigate to Traffic Management > Load Balancing > Services.
9. In the details pane, on the Action tab, click Internal Services.
10. Select nsrpcs-127.0.0.1-3008 from the list, and then click Open.
11. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
12. Select nskrpcs-127.0.0.1-3009 from the list, and then click Open.
13. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
14. Select nsrpcs-::11-3008 from the list, and then click Open.
15. On the SSL Settings tab, in the Available pane, select the certificate created in step 7, click Add, and then click OK.
16. Click OK.
17. Navigate to System > Network > RPC.
18. In the details pane, select the IP address, and click Open.
19. In the Configure RPC Node dialog box, select Secure.
20. Click OK.

Common network topologies

September 21, 2020

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

Set up a common two-arm topology

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

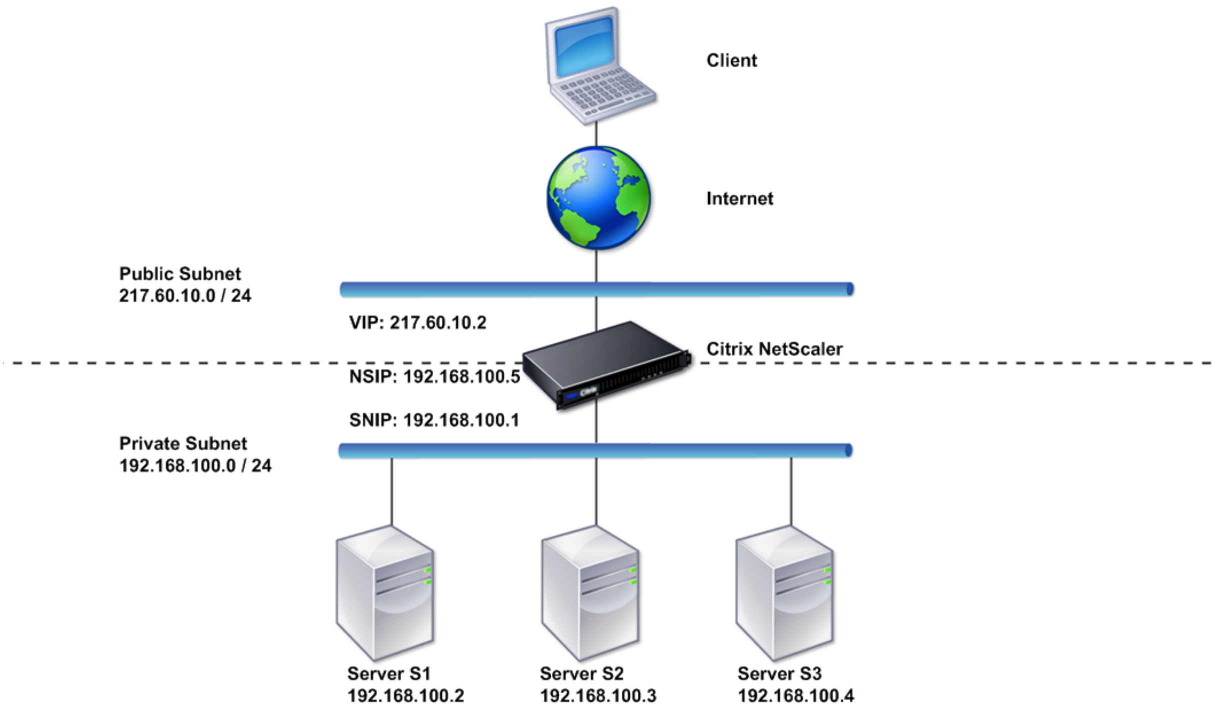
Set up a simple two-arm multiple subnet topology

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

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

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

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


Client
Internet
Public Subnet
217.60.10.0 / 24
VIP: 217.60.10.2
Citrix NetScaler
NSIP: 192.168.100.5
SNIP: 192.168.100.1
Private Subnet
192.168.100.0 / 24
Server S1
192.168.100.2
Server S2
192.168.100.3
Server S3
192.168.100.4

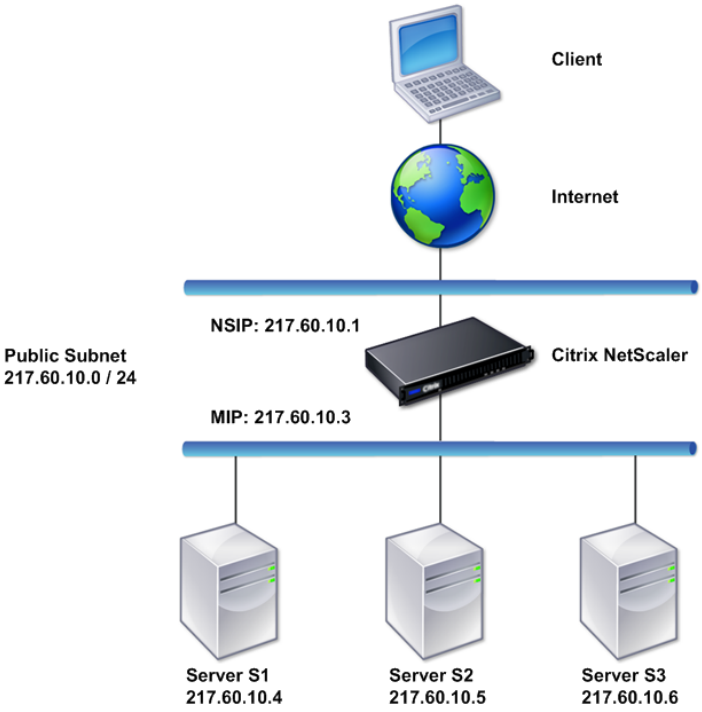
To deploy a NetScaler appliance in two-arm mode with multiple subnets, follow these steps:

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

Set up a simple two-arm transparent topology

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

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


Client
Internet
NSIP: 217.60.10.1
Public Subnet
217.60.10.0 / 24
Citrix NetScaler
MIP: 217.60.10.3
Server S1
217.60.10.4
Server S2
217.60.10.5
Server S3
217.60.10.6

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

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

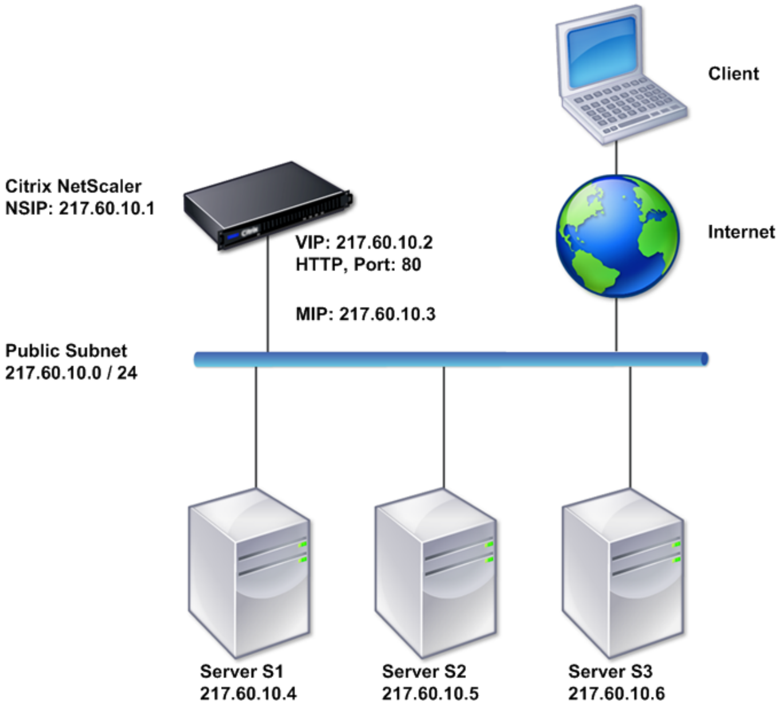
Set up common one-arm topologies

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

Set up a simple one-arm single subnet topology

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

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


Client
Citrix NetScaler
NSIP: 217.60.10.1
VIP: 217.60.10.2
HTTP, Port: 80
Internet
MIP: 217.60.10.3
Public Subnet
217.60.10.0 / 24
Server S1
217.60.10.4
Server S2
217.60.10.5
Server S3
217.60.10.6

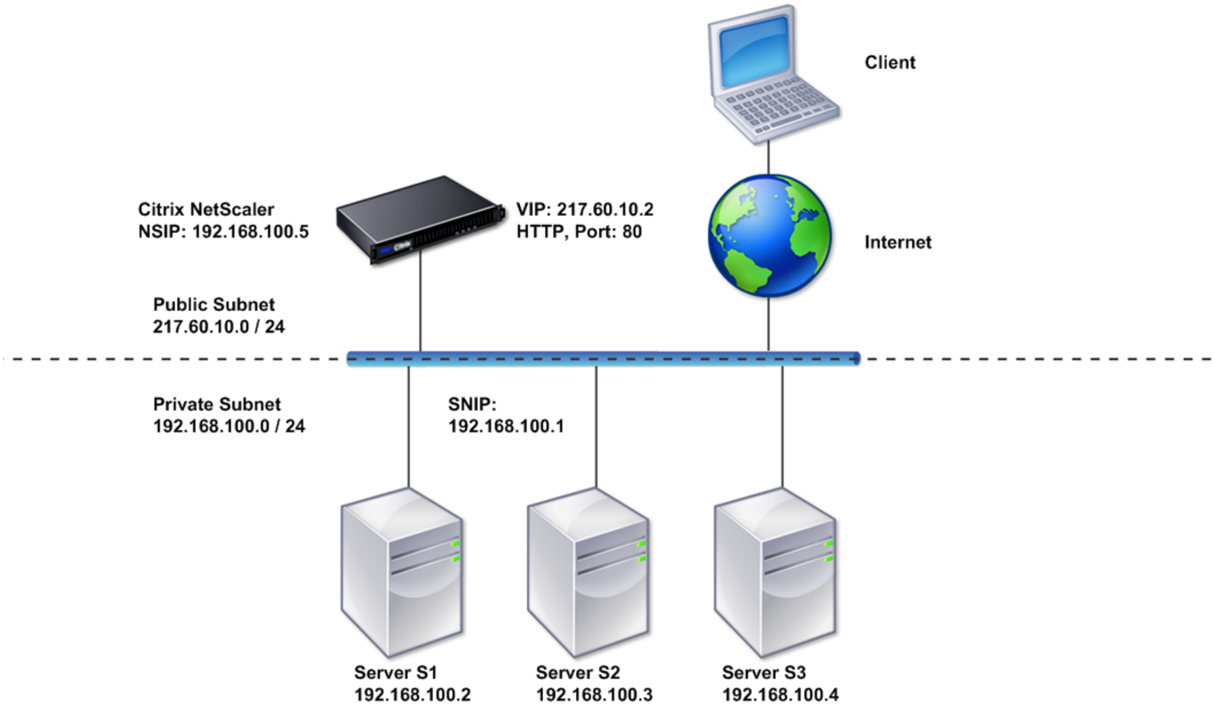
To deploy a NetScaler appliance in one-arm mode with a single subnet, follow these steps:

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

Set up a simple one-arm multiple subnet topology

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

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


Client
Citrix NetScaler
NSIP: 192.168.100.5
VIP: 217.60.10.2
HTTP, Port: 80
Internet
Public Subnet
217.60.10.0 / 24
Private Subnet
192.168.100.0 / 24
SNIP:
192.168.100.1
Server S1
192.168.100.2
Server S2
192.168.100.3
Server S3
192.168.100.4

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

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

System management settings

September 21, 2020

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

System settings

September 21, 2020

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

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

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

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

Parameter Type: HTTP Port Information

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

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

Parameter Type: Limits

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

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

Parameter Type: Client IP Insertion

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

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

Parameter Type: Cookie Version

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

Parameter Type: Requests/Responses

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

Parameter Type: Server Header Insertion

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

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

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

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

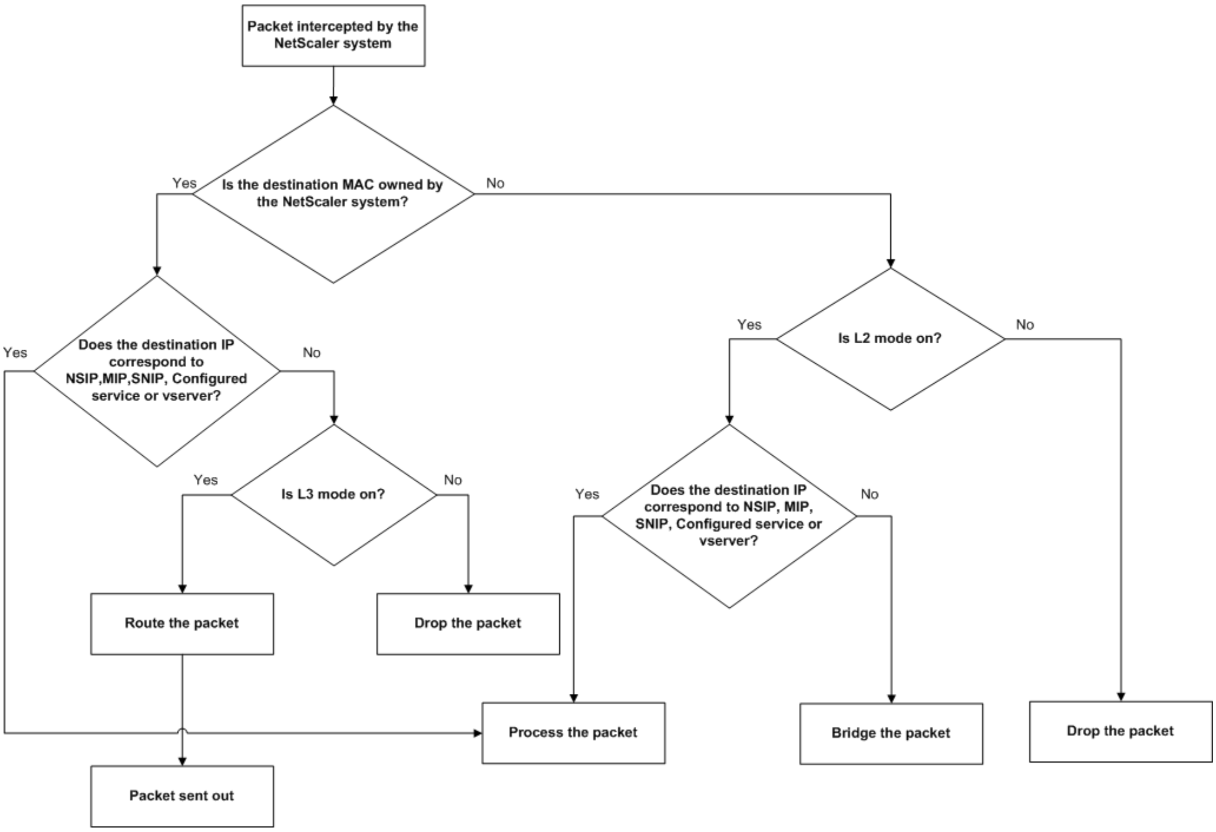
1. In the navigation pane, expand **System**, and then click **Settings**
2. In the details pane, under **Settings**, click **Change global system settings**.
3. Under **FTP Port Range,** in the **Start Port** and **End Port** text boxes, type the lowest and highest port numbers, respectively, for the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

Packet forwarding modes

September 21, 2020

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

Figure 1. Interaction between Layer 2 and Layer 3 Modes


Packet intercepted by the NetScaler system
Is the destination MAC owned by the NetScaler system?
Yes
No
Does the destination IP correspond to NSIP,MIP,SNIP, Configured service or vserver?
Yes
No
Is L3 mode on?
Yes
No
Route the packet
Drop the packet
Process the packet
Packet sent out
Is L2 mode on?
Yes
No
Does the destination IP correspond to NSIP, MIP, SNIP, Configured service or vserver?
Yes
No
Bridge the packet
Drop the packet

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

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

Enable and disable layer 2 mode

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

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

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

To enable or disable Layer 2 mode by using the CLI

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

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

  Examples

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

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

  > disable ns mode l2
  Done
  > show ns mode

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

To enable or disable Layer 2 mode by using the GUI

1. In the navigation pane, expand **System**, and then click **Settings**.
2. In the details pane, under **Modes** and **Features**, click **Configure modes**.
3. In the **Configure Modes** dialog box, to enable Layer 2 mode, select the **Layer 2 Mode** check box. To disable Layer 2 mode, clear the check box.

4. Click **OK**. The Enable/Disable Mode(s)? message appears in the details pane.
5. Click **Yes**.

Enable and disable layer 3 mode

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

To enable or disable Layer 3 mode by using the CLI

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

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

  Examples

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

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

  > disable ns mode l3
  Done
  > show ns mode
  ```

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

To enable or disable Layer 3 mode by using the GUI

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

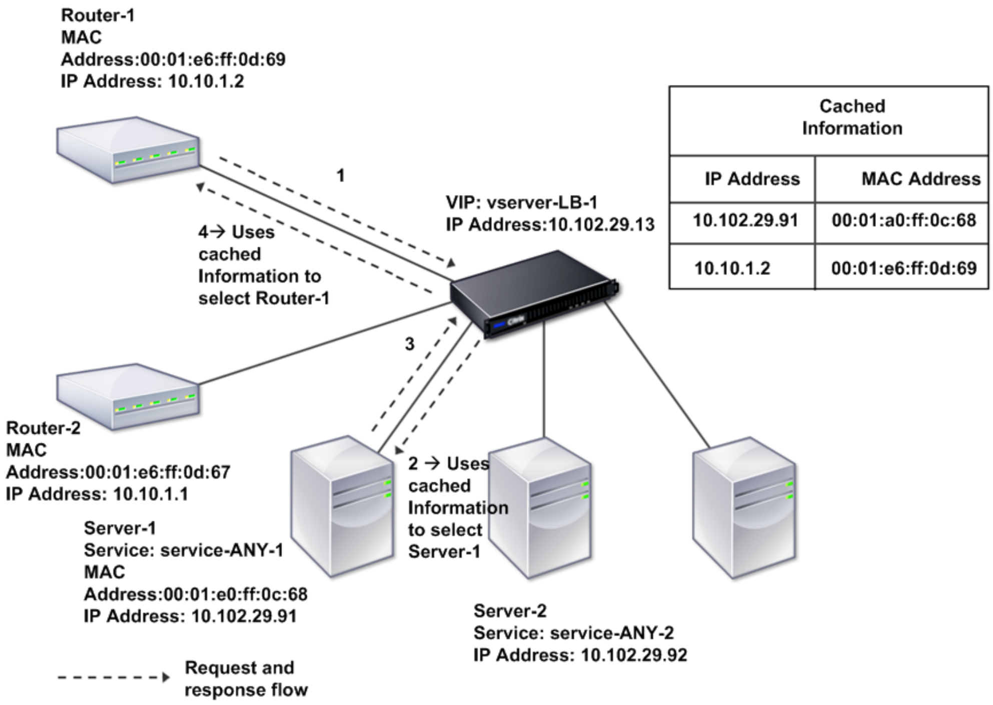
Enable and disable MAC based forwarding mode

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

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

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

Figure 2. MAC-Based Forwarding Process


Router-1
MAC
Address:00:01:e6:ff:0d:69
IP Address: 10.10.1.2
Cached
Information
IP Address
MAC Address
10.102.29.91
00:01:a0:ff:0c:68
10.10.1.2
00:01:e6:ff:0d:69
1
VIP: vserver-LB-1
IP Address:10.102.29.13
4→ Uses
cached
Information to
select Router-1
3
Router-2
MAC
Address:00:01:e6:ff:0d:67
IP Address: 10.10.1.1
2 → Uses
cached
Information
to select
Server-1
Server-1
Service: service-ANY-1
MAC
Address:00:01:e0:ff:0c:68
IP Address: 10.102.29.91
Server-2
Service: service-ANY-2
IP Address: 10.102.29.92
Request and
response flow

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

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

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

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

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

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

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

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

  "' pre codeblock

  > enable ns mode mbf
  > Done
  > show ns mode

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

  > disable ns mode mbf
  > Done
  > show ns mode

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

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

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

Network interfaces

September 21, 2020

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

Virtual LANs

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

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

The NetScaler appliance supports the following types of VLANs.

- Default VLAN

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

- Port-Based VLANs

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

- Tagged VLAN

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

Link aggregate channels

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

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

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

Clock synchronization

September 21, 2020

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

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

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

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

   restrict localhost

   restrict 127.0.0.2

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

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

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

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

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

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

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

6. Reboot the appliance to enable clock synchronization.

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

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

DNS configuration

September 21, 2020

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

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

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

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

Add a name server by using the CLI

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

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

  Example

  “‘ pre codeblock

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

  “‘

Add a name server by using the GUI

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

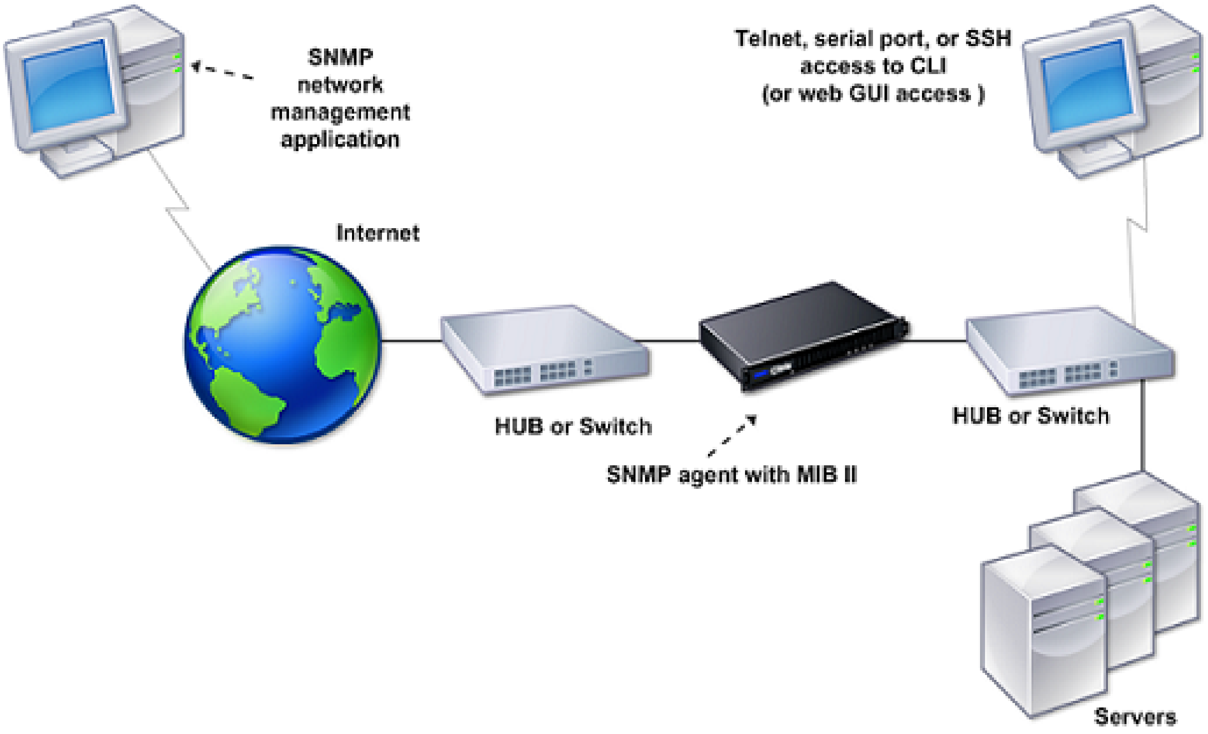
SNMP configuration

September 21, 2020

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

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

Figure 1. SNMP on the NetScaler appliance



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

The NetScaler appliance supports the following enterprise-specific MIBs:

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

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

Add SNMP managers

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

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

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

```
s********how snmp manager******** <IPAddress>
```

Example:

```
add snmp manager 10.102.29.5 -netmask 255.255.255.255`
Done
show snmp manager 10.102.29.5
10.102.29.5 255.255.255.255
Done
<!--NeedCopy-->
```

To add an SNMP manager by using the GUI:

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

Add SNMP traps listeners

After configuring the alarms, you need to specify the trap listener to which the appliance will send the trap messages. Apart from specifying parameters like IP address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

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

To add an SNMP trap listener by using the CLI

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

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

Example:

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

To add an SNMP trap listener by using the GUI

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

Configure SNMP alarms

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

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

To enable or disable an alarm by using the CLI

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

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

Example:

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

To set the severity of the alarm by using the CLI

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

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

Example:

```
set snmp alarm LOGIN-FAILURE -severity Major
Done
show snmp alarm LOGIN-FAILURE
```

```
Alarm Alarm Threshold Normal Threshold Time State Severity Logging
\----- ---------------- ----------------- ---- --------- ----------
    --------
\) LOGIN-FAILURE N/A N/A N/A ENABLED Major ENABLED
Done
<!--NeedCopy-->
```

To configure alarms by using the GUI

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

Verify configuration

September 21, 2020

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

Configuration checklist

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

Topology configuration checklist

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

The routes entered are:

____

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

____

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

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

  ____

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

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

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

Server configuration checklist

- "Keep-alive" has been enabled on all the servers.

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

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

  ____

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

  ____

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

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

____

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

  ____

Software features configuration checklist

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

  Reason for enabling or disabling:

  ____

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

  Reason for enabling or disabling:

  ____

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

  Reason for enabling or disabling:

  ____

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

  Reason for changing or not changing:

  ____

Access checklist

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

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

Firewall checklist

The following firewall requirements have been met:

- UDP 161 (SNMP)
- UDP 162 (SNMP trap)
- TCP/UDP 3010 (GUI)
- HTTP 80 (GUI)
- TCP 22 (SSH)

Load balance traffic on a NetScaler appliance

September 21, 2020

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

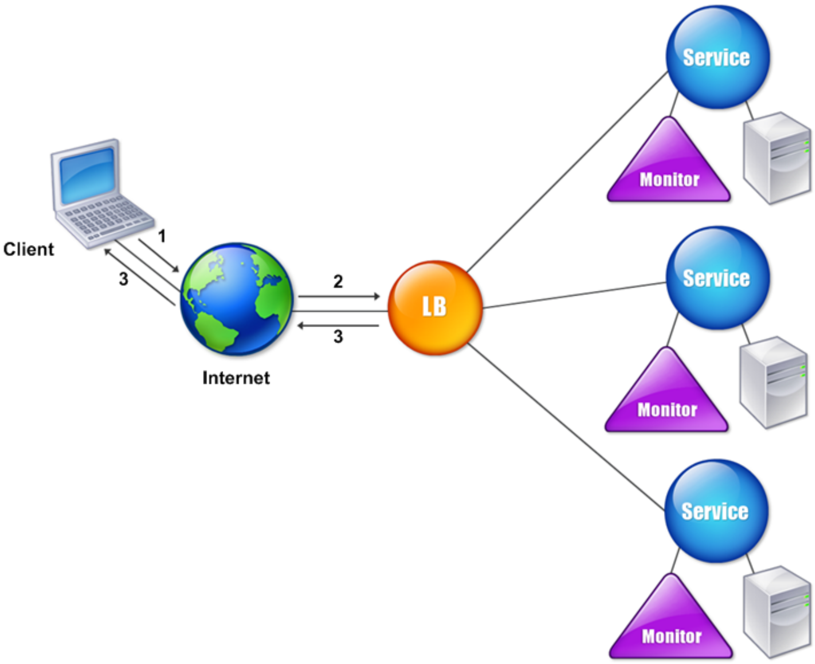
To configure load balancing, you define a virtual server to proxy multiple servers in a server farm and balance the load among them.

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

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

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

Figure 1. Load Balancing Architecture



The entities function as follows:

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

Load balancing

September 21, 2020

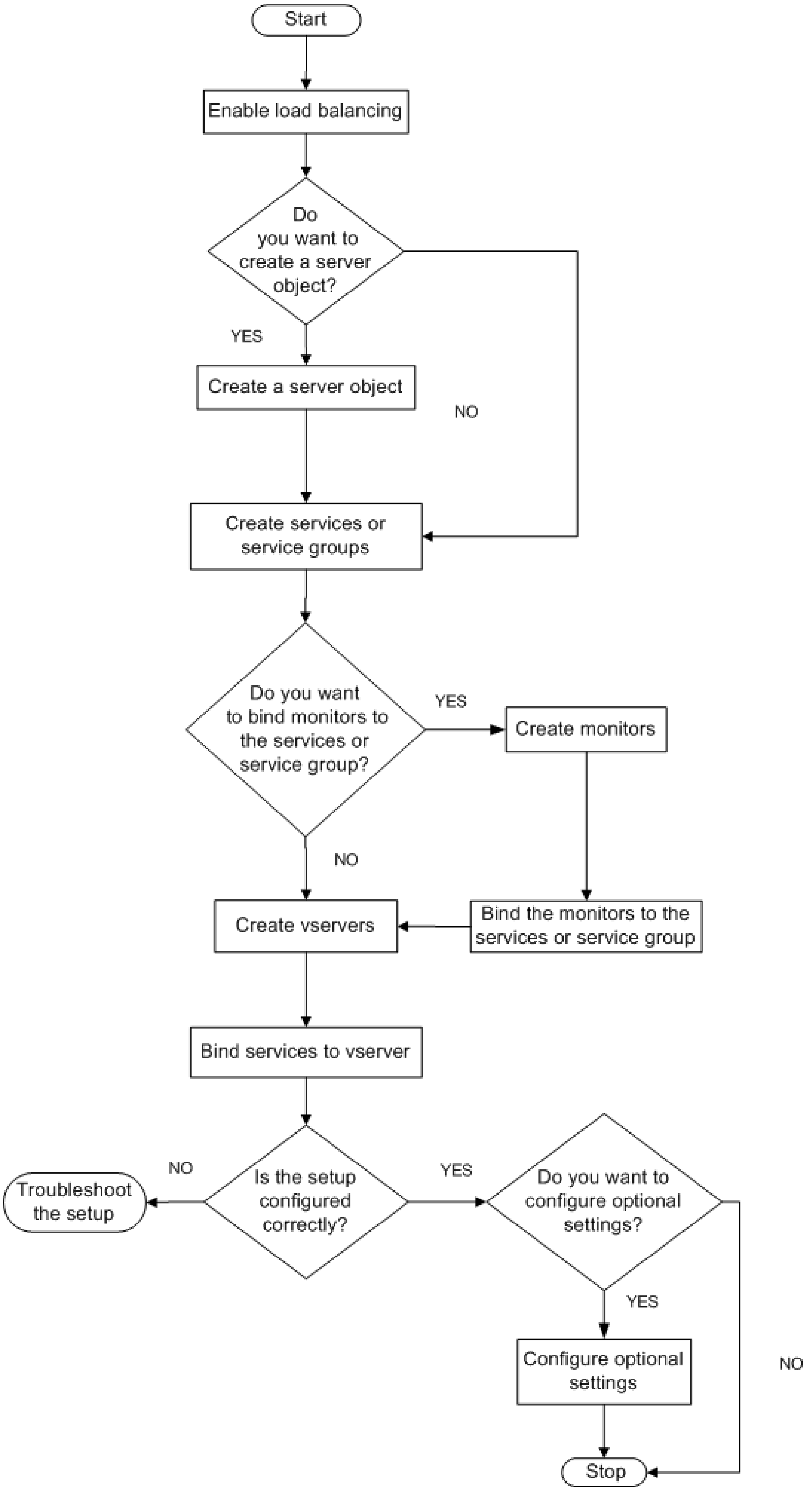
To configure load balancing, you must first create services. Then, you create virtual servers and bind the services to the virtual servers. By default, the NetScaler appliance binds a monitor to each service. After binding the services, verify your configuration by making sure that all of the settings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

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

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

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

Figure 1. Sequence of Tasks to Configure Load Balancing


Start
Enable load balancing
Do you want to create a server object?
YES
NO
Create a server object
Create services or service groups
Do you want to bind monitors to the services or service group?
YES
Create monitors
NO
Create vservers
Bind the monitors to the services or service group
Bind services to vserver
NO
Troubleshoot the setup
Is the setup configured correctly?
YES
Do you want to configure optional settings?
YES
NO
Configure optional settings
Stop

Enable load balancing

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

To enable load balancing by using the CLI

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

- enable feature lb
- show feature

  Example

  “‘ pre codeblock

  > enable feature lb
  > Done
  > show feature

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

To enable load balancing by using the GUI

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

Configure services and a virtual server

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

To implement the initial load balancing configuration by using the CLI

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

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

  Example

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

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

To implement the initial load balancing configuration by using the GUI

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

Persistence settings

September 21, 2020

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

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

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

Table 1. Limitations on Number of Simultaneous Persistent Connections

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

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

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

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

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

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

Configure persistence based on cookies

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

The NetScaler appliance inserts the cookie <NSC_XXXX>= <ServiceIP> <ServicePort>

where:

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

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

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

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

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

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

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

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

An administrator can change the HTTP cookie version.

To change the HTTP cookie version by using the CLI

At the command prompt, type;

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

Example:

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

To change the HTTP cookie version by using the GUI

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

Note: For information about the parameters, see
Configuring Persistence Based on Cookies.

To configure persistence based on cookies by using the CLI

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

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

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

Example:

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

To configure persistence based on cookies by using the GUI

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

Configure persistence based on server IDs in URLs

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

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. For more information about expressions, see "Policy Configuration and Reference."

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

Example: Payload Expression

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

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

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

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

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

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

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

Example:

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

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

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

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

Configure features to protect the load balancing configuration

September 21, 2020

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

Configure URL redirection

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

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

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

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

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

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

  Example

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

<!--NeedCopy-->
```

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

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

Configure backup virtual servers

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

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

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

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

To configure a backup virtual server by using the CLI

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

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

Example

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

```

<!--NeedCopy-->
```

To set up a backup virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the backup virtual server (for example, vserver-LB-1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Advanced tab, in the Backup Virtual Server list, select the backup virtual server (for example, vserver-LB-2, and then click OK.
4. Verify that the backup virtual server you configured appears in the Details section at the bottom of the pane.
   Note: If the primary server goes down and then comes back up, and you want the backup virtual server to function as the primary server until you explicitly reestablish the primary virtual server, select the
   Disable Primary When Down check box.

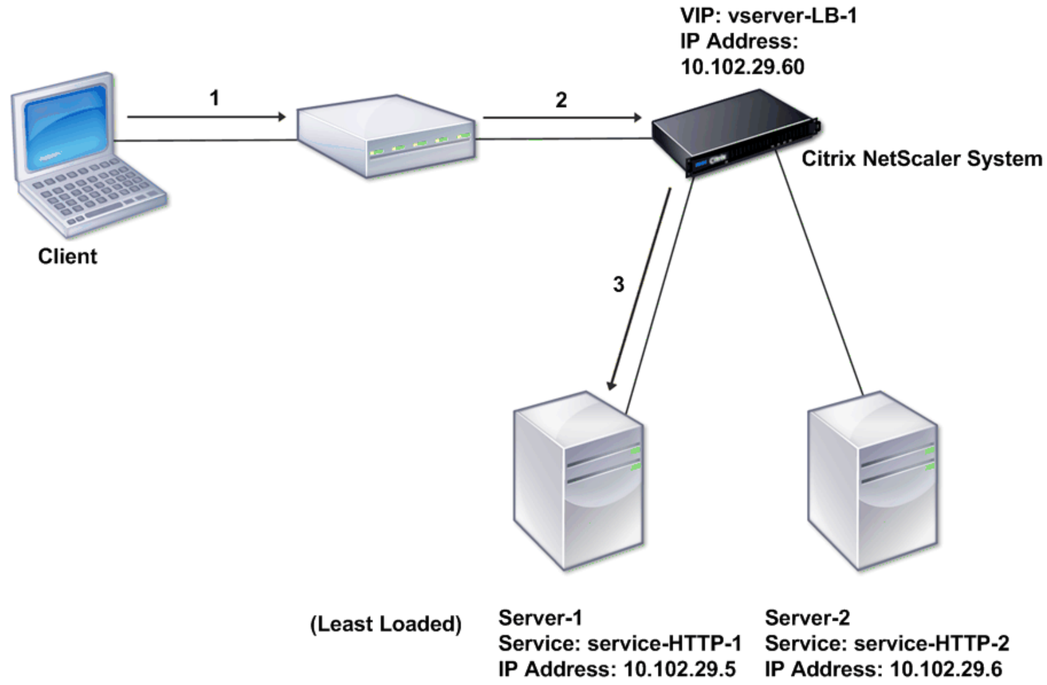
A typical load balancing scenario

September 21, 2020

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

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

Figure 1. Basic Load Balancing Topology

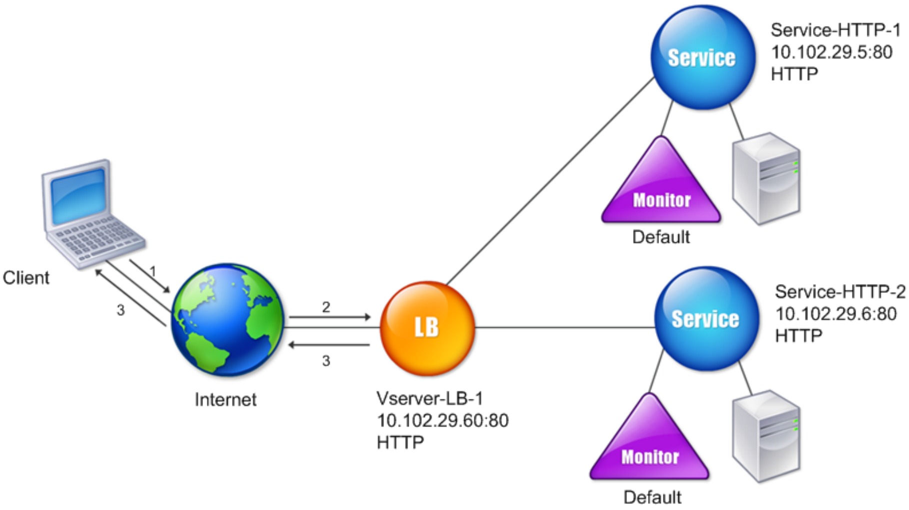

VIP: vserver-LB-1
IP Address:
10.102.29.60
1
2
Citrix NetScaler System
Client
3
(Least Loaded)
Server-1
Service: service-HTTP-1
IP Address: 10.102.29.5
Server-2
Service: service-HTTP-2
IP Address: 10.102.29.6

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

Table 1. LB Configuration Parameter Values

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

Figure 2. Load Balancing Entity Model


Service
Service-HTTP-1
10.102.29.5:80
HTTP
Monitor
Default
Client
1
3
2
3
Internet
LB
Vserver-LB-1
10.102.29.60:80
HTTP
Service
Service-HTTP-2
10.102.29.6:80
HTTP
Monitor
Default

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

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

Table 2. Initial Configuration Tasks

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

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

Table 3. Verification Tasks

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

Table 4. Customization Tasks

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

Accelerate load balanced traffic by using compression

September 21, 2020

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

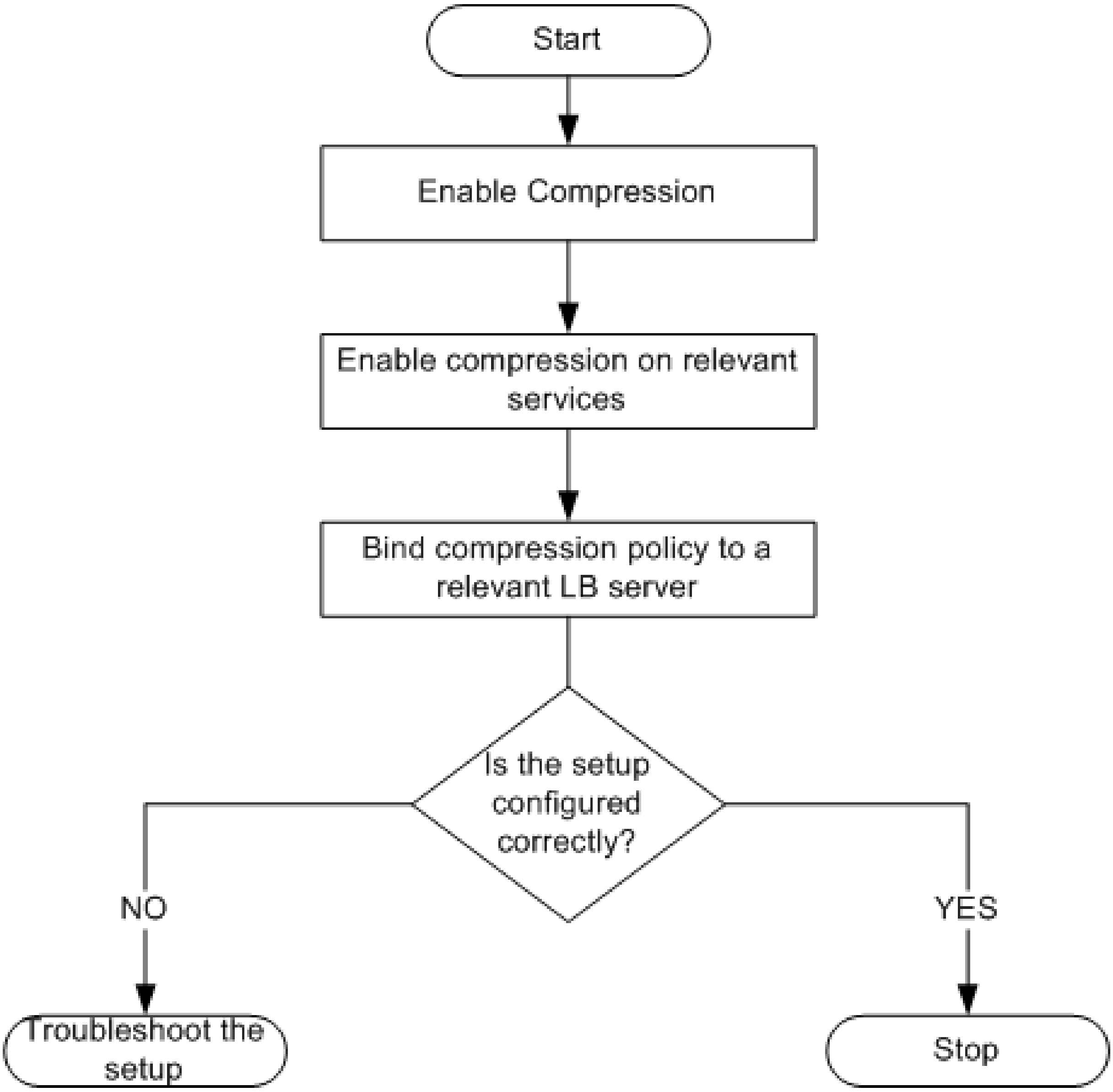
NetScaler compression is a policy-based feature. A policy filters requests and responses to identify responses to be compressed, and specifies the type of compression to apply to each response. The appliance provides several built-in policies to compress common MIME types such as text/html, text/plain, text/xml, text/css, text/rtf, application/msword, application/vnd.ms-excel, and application/vnd.ms-powerpoint. You can also create custom policies. The appliance does not compress compressed MIME types such as application/octet-stream, binary, bytes, and compressed image formats such as GIF and JPEG.

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

Compression configuration task sequence

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

Figure 1. Sequence of Tasks to Configure Compression


Start
Enable Compression
Enable compression on relevant services
Bind compression policy to a relevant LB server
Is the setup configured correctly?
NO
YES
Troubleshoot the setup
Stop

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

Enable compression

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

To enable compression by using the CLI

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

- enable ns feature CMP
- show ns feature

```
> enable ns feature CMP




Done


> show ns feature





       Feature                        Acronym              Status


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


1)     Web Logging                    WL                   ON


2)     Surge Protection               SP                   OFF


.


7) Compression Control CMP ON


8)     Priority Queuing               PQ                   OFF


.


Done
```

To enable compression by using the GUI

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

Configure services to compress data

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

To enable compression on a service by using the CLI

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

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

```
> show service SVC_HTTP1


SVC_HTTP1 (10.102.29.18:80) - HTTP


State: UP


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


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


Server Name: 10.102.29.18


Server ID : 0   Monitor Threshold : 0


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


```

```
Use Source IP: NO


Client Keepalive(CKA): NO


Access Down Service: NO


TCP Buffering(TCPB): NO


HTTP Compression(CMP): YES


Idle timeout: Client: 180 sec   Server: 360 sec


Client IP: DISABLED


Cacheable: NO


SC: OFF


SP: OFF


Down state flush: ENABLED






1)      Monitor Name: tcp-default


State: DOWN     Weight: 1


Probes: 1095    Failed [Total: 1095 Current: 1095]


```

```
Last response: Failure - TCP syn sent, reset received.


Response Time: N/A


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

To enable compression on a service by using the GUI

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

Bind a compression policy to a virtual server

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

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

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

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

-
- show lb vserver <name>

Example:

```
      bind lb vserver lbvip -policyName ns_cmp_msapp
  Done
 > show lb vserver lbvip
 lbvip (8.7.6.6:80) - HTTP Type: ADDRESS
 State: UP
 Last state change was at Thu May 28 05:37:21 2009 (+685 ms)
 Time since last state change: 19 days, 04:26:50.470
 Effective State: UP
 Client Idle Timeout: 180 sec
 Down state flush: ENABLED
 Disable Primary Vserver On Down : DISABLED
 Port Rewrite : DISABLED
 No. of Bound Services : 1 (Total) 1 (Active)
 Configured Method: LEASTCONNECTION
 Current Method: Round Robin, Reason: Bound service's state changed to UP
 Mode: IP
 Persistence: NONE
 Vserver IP and Port insertion: OFF
 Push: DISABLED Push VServer:
 Push Multi Clients: NO
 Push Label Rule:

 Bound Service Groups:
 1) Group Name: Service-Group-1

 1) Service-Group-1 (10.102.29.252: 80) - HTTP State: UP Weight: 1

 1) Policy : ns_cmp_msapp Priority:0
 Done
<!--NeedCopy-->
```

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

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

  - To unbind a compression policy, click the name of the policy you want to unbind from the virtual server, and then click Unbind Policy.
5. Click OK.

Secure load balanced traffic by using SSL

September 21, 2020

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

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

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

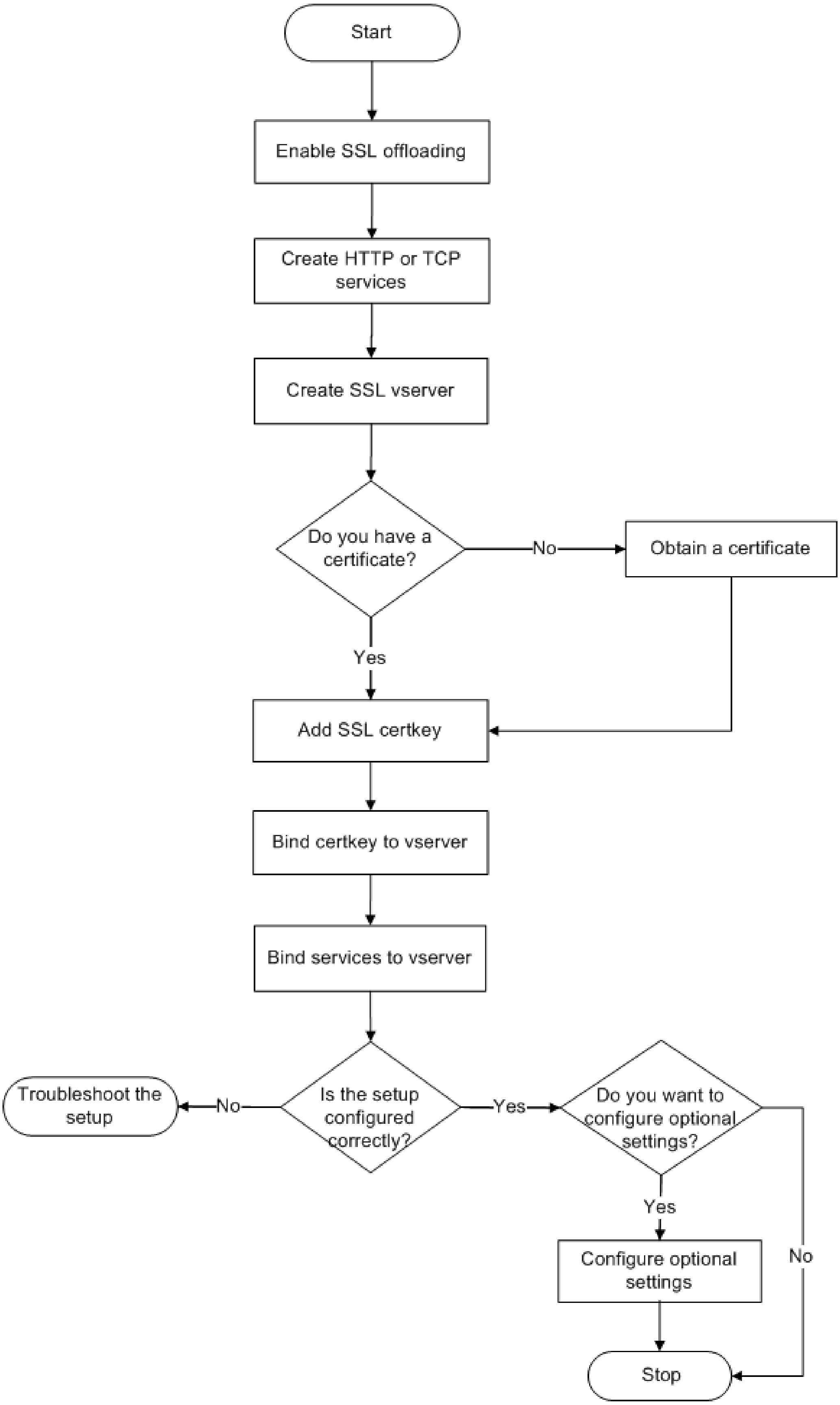
SSL configuration task sequence

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

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

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

Figure 1. Sequence of Tasks to Configure SSL Offloading


Start
Enable SSL offloading
Create HTTP or TCP services
Create SSL vserver
Do you have a certificate?
No
Obtain a certificate
Yes
Add SSL certkey
Bind certkey to vserver
Bind services to vserver
Troubleshoot the setup
No
Is the setup configured correctly?
Yes
Do you want to configure optional settings?
Yes
Configure optional settings
No
Stop

Enable SSL offload

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

Eenable SSL by using the CLI

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

- enable ns feature SSL
- show ns feature

```
> enable ns feature ssl




Done


> show ns feature


Feature Acronym Status


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


1) Web Logging WL ON


2) SurgeProtection SP OFF


3) Load Balancing LB ON . . .


 9) SSL Offloading SSL ON


10) Global Server Load Balancing GSLB ON . .


```

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

Enable SSL by using the GUI

Follow thesee steps:

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

Create HTTP services

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

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

Add an HTTP service by using the CLI

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

add service <name> (<IP>	<serverName>) <serviceType> <port>

-
- show service <name>

Example:

```
> add service SVC_HTTP1 10.102.29.18 HTTP 80


 Done


> show service SVC_HTTP1


```

```
        SVC_HTTP1 (10.102.29.18:80) - HTTP


        State: UP


        Last state change was at Wed Jul 15 06:13:05 2009


        Time since last state change: 0 days, 00:00:15.350


        Server Name: 10.102.29.18


        Server ID : 0   Monitor Threshold : 0


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


        Use Source IP: NO


        Client Keepalive(CKA): NO


        Access Down Service: NO


        TCP Buffering(TCPB): NO


        HTTP Compression(CMP): YES


        Idle timeout: Client: 180 sec   Server: 360 sec


        Client IP: DISABLED


        Cacheable: NO


        SC: OFF


        SP: OFF


        Down state flush: ENABLED





1)      Monitor Name: tcp-default


                State: UP       Weight: 1


                Probes: 4       Failed [Total: 0 Current: 0]


                Last response: Success - TCP syn+ack received.


                Response Time: N/A


 Done



<!--NeedCopy-->
```

Add an HTTP service by using the GUI

Follow these steps:

1. Navigate to Traffic Management > SSL Offload > Services.
2. In details pane, click Add.
3. In the Create Service dialog box, in the Service Name, Server, and Port text boxes, type the name of the service, IP address, and port (for example, SVC_HTTP1, 10.102.29.18, and 80).
4. In the Protocol list, select the type of the service (for example, HTTP).
5. Click Create, and then click Close. The HTTP service you configured appears in the Services

page.

6. Verify that the parameters you configured are correctly configured by selecting the service and viewing the Details section at the bottom of the pane.

Add an SSL based virtual server

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

Add an SSL-based virtual server by using the CLI

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

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

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

Example:

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




  Done


  > show lb vserver vserver-SSL-1


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


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


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


```

```
    Effective State: DOWN Client Idle Timeout: 180 sec


    Down state flush: ENABLED


    Disable Primary Vserver On Down : DISABLED


    No. of Bound Services : 0 (Total) 0 (Active)


    Configured Method: LEASTCONNECTION Mode: IP


    Persistence: NONE


    Vserver IP and Port insertion: OFF


    Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:
        Done
<!--NeedCopy-->
```

Add an SSL-based virtual server by using the GUI

Follow these steps:

1. Navigate to Traffic Management > SSL Offload > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (SSL Offload) dialog box, in the Name, IP Address, and Port text boxes, type the name of the virtual server, IP address, and port (for example, Vserver-SSL-1, 10.102.29.50, and 443).
4. In the Protocol list, select the type of the virtual server, for example, SSL.
5. Click Create, and then click Close.
6. Verify that the parameters you configured are correctly configured by selecting the virtual server and viewing the Details section at the bottom of the pane. The virtual server is marked as DOWN because a certificate-key pair and services have not been bound to it.

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

Bind services to the SSL virtual server

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

Data transfer between the appliance and the servers can be encrypted or in clear text. If the data transfer between the appliance and the servers is encrypted, the entire transaction is secure from end to end. For more information about configuring the system for end-to-end security, see [sSSL offload and acceleration.

Bind a service to a virtual server by using the CLI

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

- bind lb vserver <name> <serviceName>
- show lb vserver <name>

Example:

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




 Done


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


 ADDRESS State: DOWN[Certkey not bound]


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


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


 Effective State: DOWN Client Idle


 Timeout: 180 sec

```

```

  Down state flush: ENABLED Disable Primary Vserver On Down :


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


  Configured Method: LEASTCONNECTION Mode: IP Persistence: NONE Vserver IP and


  Port insertion: OFF Push: DISABLED Push VServer: Push Multi Clients: NO Push Label Rule:





  1) SVC_HTTP1 (10.102.29.18: 80) - HTTP


  State: DOWN Weight: 1


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

Bind a service to a virtual server by using the GUI

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

Add a certificate key pair

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

SSL certificate. The appliance supports RSA/DSA certificates of up to 4096 bits.

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

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

Add a certificate key pair by using the CLI

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

- add ssl certKey <certkeyName> -cert <string> [-key <string>]
- show sslcertkey <name>

Example:

```
> add ssl certKey CertKey-SSL-1 -cert ns-root.cert -key ns-root.key




 Done


> show sslcertkey CertKey-SSL-1


    Name: CertKey-SSL-1 Status: Valid,


    Days to expiration:4811 Version: 3


    Serial Number: 00 Signature Algorithm: md5WithRSAEncryption Issuer: C=US,ST=California,L=San


    Jose,O=Citrix ANG,OU=NS Internal,CN=de fault


    Validity Not Before: Oct 6 06:52:07 2006 GMT Not After : Aug 17 21:26:47 2022 GMT


    Subject: C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,
        CN=d efault Public Key


    Algorithm: rsaEncryption Public Key


    size: 1024


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

Add a certificate key pair by using the GUI

Follow these steps:

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

Bind an SSL certificate key pair to the virtual server

After you have paired an SSL certificate with its corresponding key, you must bind the certificate key pair to the SSL virtual server so that it can be used for SSL processing. Secure sessions require establishing a connection between the client computer and an SSL-based virtual server on the appliance. SSL processing is then carried out on the incoming traffic at the virtual server. Therefore, before enabling the SSL virtual server on the appliance, you need to bind a valid SSL certificate to the SSL virtual server.

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

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

- bind ssl vserver <vServerName> -certkeyName <string>
- show ssl vserver <name>

Example:

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




Done


> show ssl vserver Vserver-SSL-1





     Advanced SSL configuration for VServer Vserver-SSL-1:


     DH: DISABLED


     Ephemeral RSA: ENABLED Refresh Count: 0


     Session Reuse: ENABLED Timeout: 120 seconds


     Cipher Redirect: ENABLED


     SSLv2 Redirect: ENABLED


     ClearText Port: 0


     Client Auth: DISABLED


     SSL Redirect: DISABLED


     Non FIPS Ciphers: DISABLED


     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED






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


1) Cipher Name: DEFAULT


   Description: Predefined Cipher Alias


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

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

Follow these steps:

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

Configure support for Outlook web access

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

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

To configure OWA support, do the following:

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

Create an SSL action to enable OWA support

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

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

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

- add ssl action <name> -OWASupport ENABLED
- show SSL action <name>

Example:

```
> add ssl action Action-SSL-OWA -OWASupport enabled




Done


> show SSL action Action-SSL-OWA


Name: Action-SSL-OWA


Data Insertion Action: OWA

```

```

Support: ENABLED


Done
```

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

Follow these steps:

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

Create SSL policies

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

Create an SSL policy by using the CLI

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

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

Example:

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

Done

> show ssl policy Policy-SSL-1

Name: Policy-SSL-1 Rule: ns_true

Action: Action-SSL-OWA Hits: 0

```

```
Policy is bound to following entities

1) PRIORITY : 0

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

Create an SSL policy by using the GUI

Follow these steps:

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, click Add.
3. In the Create SSL Policy dialog box, in the Name text box, type the name of the SSL Policy (for example, Policy-SSL-1).
4. In Request Action, select the configured SSL action that you want to associate with this policy (for example, Action-SSL-OWA). The ns_true general expression applies the policy to all successful SSL handshake traffic. However, if you need to filter specific responses, you can create policies with a higher level of detail. For more information about configuring granular policy expressions, see [SSL actions and policies](/en-us/netscaler/11-1/ssl/ssl-actions-and-policies.html0.
5. In Named Expressions, choose the built-in general expression ns_true and click Add Expression. The expression ns_true now appears in the Expression text box.
6. Click Create, and then click Close.
7. Verify that the policy is correctly configured by selecting the policy and viewing the Details section at the bottom of the pane.

Bind the SSL policy to the SSL virtual server

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

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

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

- bind ssl vserver <vServerName> -policyName <string>
- show ssl vserver <name>

Example:

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

 Done

> show ssl vserver Vserver-SSL-1

Advanced SSL configuration for VServer Vserver-SSL-1:

DH: DISABLED

Ephemeral RSA: ENABLED

Refresh Count: 0

Session Reuse: ENABLED

Timeout: 120 seconds

Cipher Redirect: ENABLED

SSLv2 Redirect: ENABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

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

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

1) Cipher Name: DEFAULT Description: Predefined Cipher Alias

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

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

Follow these steps:

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

Features at a glance

September 21, 2020

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

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

Application switching and traffic management features

September 21, 2020

Below are the application switching and traffic management features.

SSL Offloading

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

For more information, see SSL offload and acceleration.

Access Control Lists

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

For more information, see Access Control List.

Load Balancing

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

For more information, see Load Balancing.

Traffic Domains

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

For more information, see Traffic Domains.

Network Address Translation

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

The NetScaler appliance supports the following types of network address translation:

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

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

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

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

For more information, see
Configuring Network Address Translation.

Multipath TCP Support

NetScaler appliances support Multipath TCP (MPTCP). MPTCP is a TCP/IP protocol extension that identifies and uses multiple paths available between hosts to maintain the TCP session. You must enable MPTCP on a TCP profile and bind it to a virtual server. When MPTCP is enabled, the virtual server functions as an MPTCP gateway and converts MPTCP connections with the clients to TCP connections that it maintains with the servers.

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

Content Switching

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

For more information, see Content Switching.

Global Server Load Balancing (GSLB)

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

For more information, see Global Server Load Balancing.

Dynamic Routing

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

For more information, see Configuring Dynamic Routes.

Link Load Balancing

Load balances multiple WAN links and provides link failover, further optimizing network performance and ensuring business continuity. Ensures that network connections remain highly available, by applying intelligent traffic control and health checks to distribute traffic efficiently across upstream routers. Identifies the best WAN link to route both incoming and outbound traffic based on policies and network conditions, and protects applications against WAN or Internet link failure by providing rapid fault detection and failover.

```
For more information, see [Link Load Balancing](/en-us/netscaler/11-1/
    link-load-balancing.html).
```

TCP Optimization

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

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

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

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

CloudBridge Connector

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

For more information, see CloudBridge.

DataStream

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

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

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

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

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

For more information, see DataStream.

Application acceleration features

September 21, 2020

- AppCompress

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

- Cache Redirection

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

  For more information, see "Cache Redirection."

- AppCache

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

For more information, see "Integrated Caching."

- TCP Buffering

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

Application security and firewall features

September 21, 2020

Below are the secuirty and firewarall features.

Denial of service (DoS) attack defense

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

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

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

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

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

Content Filtering

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

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

For more information, see Content Filtering.

Responder

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

For more information, see Responder.

Rewrite

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

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

For more information, see Rewrite.

Priority Queuing

Prioritizes user requests to ensure that the most important traffic is serviced first during surges in request volume. You can establish priority based on request URLs, cookies, or a variety of other factors. The appliance places requests in a three-tier queue based on their configured priority, enabling business-critical transactions to flow smoothly even during surges or site attacks.

For more information, see Priority Queuing.

Surge Protection

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

For more information, see Surge Protection.

Citrix Gateway

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

For more information, see [Citrix Gateway](https://docs.citrix.com/en-us/netscaler-gateway.html).
```

Application Firewall

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

For more information, see Application Firewall.

Application visibility feature

September 21, 2020

- NetScaler Insight Center

NetScaler Insight Center is a high performance collector that provides end-to-end user experience visibility across Web and HDX (ICA) traffic. It collects HTTP and ICA AppFlow records generated by NetScaler ADC appliances and populates analytical reports covering Layer 3 to Layer 7 statistics. NetScaler Insight Center provides in-depth analysis for the last five minutes of real-time data, and for historical data collected for the last one hour, one day, one week, and one month.
HDX (ICA) analytic dashboard enables you to drill down from HDX Users, Applications, Desktops, and even from gateway-level information. Similarly, HTTP analytics provide a bird's eye view of Web Applications, URLs Accessed, Client IP Addresses and Server IP Addresses, and other dashboards. The administrator can drill down and identify the pain points from any of these dashboards, as appropriate for the use case.

- EdgeSight for NetScaler

  Support for application performance monitoring based on end user experience. This solution leverages the HTML injection feature to obtain various time values, which are used by EdgeSight server for analysis and report generation. EdgeSight for NetScaler provides a way to monitor the performance benefits of a NetScaler and determine potential bottlenecks in a network.

  For more information, see "EdgeSight Monitoring for NetScaler."

- Enhanced Application Visibility Using AppFlow

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

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

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

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

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

  For more information, see "AppFlow."

- Stream Analytics

The performance of your web site or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

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

For more information, see “Action Analytics.”

Solutions for Telecom Service Providers

September 21, 2020

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

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

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

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

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

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

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

Large Scale NAT

September 21, 2020

> **Note**
>
> This feature is available with a NetScaler Advanced or Premium edition license.

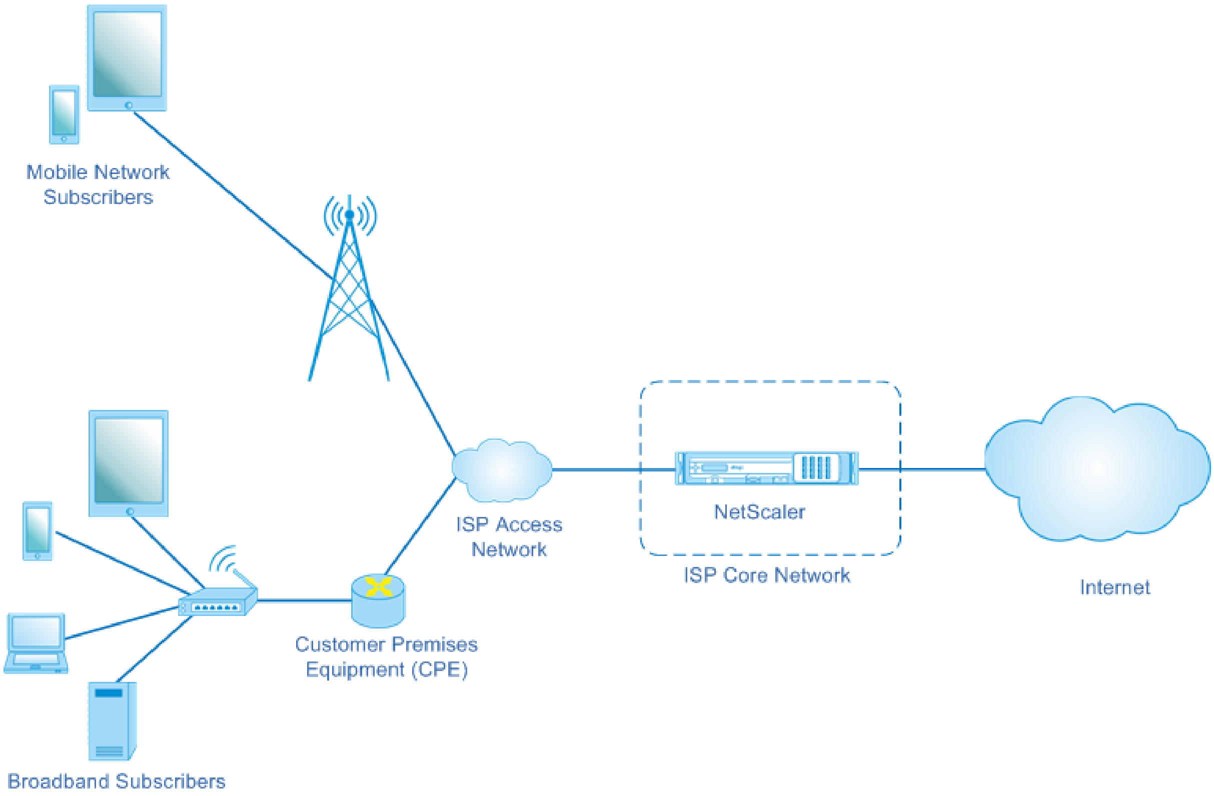
The Internet's phenomenal growth has resulted in a shortage of public IPv4 addresses. Large Scale NAT (LSN/CGNAT) provides a solution to this issue, maximizing the use of available public IPv4 addresses by sharing a few public IPv4 addresses among a large pool of Internet users.

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

LSN Architecture

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

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



Large Scale NAT

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

LSN Features Supported on NetScaler appliance

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

NAT Resource Allocation

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

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

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

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

IP Pooling

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

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

Reusing LSN Mappings

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

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

LSN Filtering

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

1. **Endpoint Independent.** The NetScaler appliance filters out only those packets that are not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x), regardless of the external host IP address and port source (Z:z). The NetScaler appliance forwards any packets destined to X:x. In other words, sending packets from the subscriber to any external IP address is sufficient to allow packets from any external host to the subscriber. This type of filtering is useful

for proper functioning of VOIP and peer-to-peer applications.

2. **Address dependent.** The NetScaler appliance filters out packets not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x). In addition, the appliance filters out packets from external host IP address and port (Y:y) destined for N:n if the subscriber has not previously sent packets to Y:anyport (external port independent). In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external host's IP address.
3. **Address port dependent.** The NetScaler appliance filters out packets not destined to NAT IP:port (N:n), which represents subscriber IP:port (X:x). In addition, the appliance filters out packets from external host IP address and port (Y:y) destined for N:n if the subscriber has not previously sent packets to Y:y. In other words, receiving packets from a specific external host requires that the subscriber first send packets to that specific external IP address and port.

Quotas

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

- **Port quota.** The NetScaler appliance can limit the LSN NAT ports to be used at a time by each subscriber for a specified protocol. For example, you could limit each subscriber to a maximum of 500 TCP NAT ports. When the LSN NAT mappings for a subscriber reach the limit, the NetScaler appliance does not allocate additional NAT ports of the specified protocol to that subscriber.
- **Subscriber Session Limit.** The number of concurrent session for a subscriber can be more than it port quota. The NetScaler appliance can limit the LSN sessions allowed for each subscriber for a specified protocol. When the number of LSN sessions reaches the limit for a subscriber, the NetScaler appliance does not allow the subscriber to open additional sessions of the specified protocol.
- **Group Session Limit.** The NetScaler appliance can limit the total number of LSN sessions allowed for a subscriber group for a specified protocol. When the total number of LSN sessions reaches the limit for a group for a specified protocol, the NetScaler appliance does not allow any subscriber of the group to open additional sessions of the specified protocol. For example, You limit a group to a maximum of 10000 UDP sessions. When the total number of UDP sessions for this group reaches 10000, the NetScaler appliance does not allow any subscriber of the group to open additional UDP sessions.

Application Layer Gateways

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

The NetScaler appliance supports ALG for the following protocols:

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

Hairpin Support

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

Points to Consider before Configuring LSN

September 21, 2020

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

- Make sure that you understand the different components of Large Scale NAT, described in RFCs 6888, 5382, 5508, and 4787.
- Endpoint independent mapping (EIM) and endpoint independent filtering (EIF) are disabled by default. These options must be enabled for proper functioning of VoIP and peer-to-peer (P2P) applications.
- **Logging LSN**: Following are the consideration points for logging LSN information:
  - Citrix recommends logging the LSN information on external log servers instead of on the NetScaler appliance. Logging on external servers facilitates optimal performance when the appliance creates large numbers of LSN log entries (in order of millions).
  - Citrix recomends using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.
  - The following limitations apply to SYSLOG over TCP:
    - The Syslog over TCP solution does not provide authentication, integrity check, and privacy.
    - The NetScaler appliance relies on the TCP protocol to provide confirmation of SYSLOG message delivery to external log servers.

- **High Availability**: Following are the consideration points for high availability of NetScaler appliances for LSN:
  - Citrix recommends configuring the LSN feature in a high availability deployment of two NetScaler appliances for uninterrupted and seamless operation of all LSN sessions.
  - In a high availability deployment, Citrix recommends:
    - Setting the SYNC VLAN parameter for dedicating a VLAN for all HA related communication.
    - Synchronizing the symmetric RSS key of the primary node to the secondary node for stateful synchronization of a large number of LSN mappings and sessions.
    - Binding the subnet of LSN IP addresses to a VLAN to avoid flooding of GARP broadcasts on all VLANs after a failover.
  - In a high availability deployment of NetScaler appliances, ALG-related sessions are not mirrored to the secondary appliance.
- **Application Layer Gateways (ALGs)**: Following are the consideration points related for ALGs on a NetScaler appliance:
  - The following are not supported for SIP ALG:
    - Multicast IP addresses
    - Encrypted SDP
    - SIP messages over TLS
    - FQDN translation in SIP messages
    - Authentication of SIP messages
    - Traffic domains, admin partitions, andNetScaler clusters.
    - SIP messages with multipart bodies.
  - The following are not supported for RTSP ALG:
    - Multicast RTSP sessions
    - RTSP session over UDP
    - NetScaler traffic domains, admin partitions, andNetScaler clusters
  - The NetScaler appliance does not support ALG for the IPSec protocol.
- If you disable the LSN feature when some LSN sessions exist on the NetScaler appliance, these sessions continue to exist for the duration of the configured timeout interval.
- LSN takes precedence over RNAT. If a packet from a specified LSN subscriber also matches a RNAT rule, the packet is translated according to the LSN configuration.
- Forwarding of packets related only to the LSN sessions is based on the NetScaler appliance's routing table.
- Unlike with subnet IP addresses, selection of an LSN NAT IP address for a subscriber's connection is not based on the routing entry for the destination IP address.
- For inbound packets, static LSN mappings take precedence over dynamic LSN mappings.
- For outbound packets, LSN application profiles take precedence over static mapping.
- When a large number of LSN sessions (> 1 million) exist on the NetScaler appliance, Citrix recom-

mends displaying selected LSN sessions instead of all of them. In the command line interface or the configuration utility, use the selection parameters for showing LSN session operation.

- LSN is not supported in a NetScaler cluster.
- To reduce the amount of active memory allocated to the LSN feature, you must warm restart the NetScaler appliance after changing the configured-memory setting. Without a warm restart, you can only increase the amount of active memory.

Configuration Steps for LSN

September 21, 2020

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

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

transport profile for that protocol.

5. **(Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it.** An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The configuration utility combines these two operations on a single screen.
6. **Create an LSN Group and bind LSN pools, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group.** An LSN group is an entity consisting of an LSN client, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. For Dynamic NAT, an LSN pool can be bound to multiple LSN groups. For Deterministic NAT, pools bound to an LSN group cannot be bound to other LSN groups. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

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

LSN entities and bindings	Limit
LSN clients	1024
LSN pools	128
LSN groups	1024
Subscriber networks that can be bound to an LSN client	64
Extended ACLs that can be bound to an LSN client	1024

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

Configuration Using the Command Line Interface

To create an LSN client by using the command line interface

At the command prompt, type:

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn client
<!--NeedCopy-->
```

To create an LSN pool by using the command line interface

At the command prompt, type:

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

show lsn pool
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

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

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

At the command prompt, type:

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

show lsn transportprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn appsprofile
```

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

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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To create an LSN group by using the command line interface

At the command prompt, type:

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

show lsn group
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn group
<!--NeedCopy-->
```

Configuration Using the Configuration Utility

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

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

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

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

To configure an LSN transport profile by using the configuration utility

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

To configure an LSN application profile by using the configuration utility

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

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

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn client
  - clientname

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

    This is a mandatory argument. Maximum Length: 127

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn client
  - clientname

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

  This is a mandatory argument. Maximum Length: 127

  - network

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

  - netmask

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

    Default value: 255.255.255.255

  - td

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

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

    Default value: 0

    Minimum value: 0

    Maximum value: 4094

  - aclname

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn pool
  - poolname

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

    This is a mandatory argument. Maximum Length: 127

  - nattype

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

Available options function as follows:

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

      Possible values: DYNAMIC, DETERMINISTIC

      Default value: DYNAMIC

- portblockallocation

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

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

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- portreallocetimeout

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

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

  In these cases, ports are immediately reallocated.

  Default value: 0

  Maximum value: 600

- maxPortReallocTmq

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

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

  Default value: 65536

  Maximum value: 65536

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn pool
  - poolname

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

    This is a mandatory argument. Maximum Length: 127

  - lsnip

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

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn transportprofile
    - transportprofilename

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

      This is a mandatory argument. Maximum Length: 127
    - transportprotocol

      Protocol for which to set the LSN transport profile parameters.

      This is a mandatory argument.

      Possible values: TCP, UDP, ICMP
    - sessiontimeout

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

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

      Default value: 120

      Minimum value: 60
    - finrsttimeout

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

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

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

Default value: 30

- portquota

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

  Default value: 0

  Minimum value: 0

  Maximum value: 65535

- sessionquota

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

  Default value: 0

  Minimum value: 0

  Maximum value: 65535

- portpreserveparity

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- portpreserverange

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

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

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

    Possible values: ENABLED, DISABLED

    Default value: DISABLED

  - syncheck

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

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

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

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

    Possible values: ENABLED, DISABLED

    Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn appsprofile
  - appsprofilename

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

    This is a mandatory argument. Maximum Length: 127

  - transportprotocol

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

    This is a mandatory argument.

Possible values: TCP, UDP, ICMP

- ippooling

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

  Available options function as follows:

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

  This parameter is applicable to dynamic NAT allocation only.

  Possible values: PAIRED, RANDOM

  Default value: RANDOM

- mapping

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

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

  Available options function as follows:

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

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

  Default value: ADDRESS-PORT-DEPENDENT

- filtering

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

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

Available options function as follows:

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

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

Default value: ADDRESS-PORT-DEPENDENT

- tcpproxy

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

  Possible values: ENABLED, DISABLED

  Default value: DISABLED

- td

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

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

  Default value: 65535

  Maximum value: 65535

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn appsprofile
    - appsprofilename

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

      This is a mandatory argument. Maximum Length: 127
    - lsnport

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

Parameter Descriptions (of commands listed in the CLI procedure)

- add lsn group
    - groupname

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

      This is a mandatory argument. Maximum Length: 127
    - clientname

      Name of the LSN client entity to be associated with the LSN group. You can associate only one LSN client entity with an LSN group. You cannot remove this association or replace with another LSN client entity once the LSN group is created.

      This is a mandatory argument. Maximum Length: 127
    - nattype

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

      Available options function as follows:

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

  Possible values: DYNAMIC, DETERMINISTIC

  Default value: DYNAMIC

- portblocksize

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

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

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

- logging

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

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

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

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

Possible values: ENABLED, DISABLED

Default value: DISABLED

- sessionLogging

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

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

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

Possible values: ENABLED, DISABLED

Default value: DISABLED

  - sessionSync

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

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

    Possible values: ENABLED, DISABLED

    Default value: ENABLED

  - snmptraplimit

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

    Default value: 100

    Minimum value: 0

    Maximum value: 10000

  - ftp

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

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

    Possible values: ENABLED, DISABLED

    Default value: ENABLED

Parameter Descriptions (of commands listed in the CLI procedure)

- bind lsn group
  - groupname

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

This is a mandatory argument. Maximum Length: 127

- poolname

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

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

- transportprofilename

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

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

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

- appsprofilename

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

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

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

Sample LSN Configurations

September 21, 2020

The following are examples of configuring LSN through command line interface.

Create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and default settings:

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

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

Create an LSN configuration with an extended ACL for identifying LSN subscribers:

```
add ns acl LSN-ACL-2 ALLOW -srcIP 192.0.2.10-192.0.2.20

Done

apply acls

Done

add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2  - aclname LSN-ACL-2

Done

add lsn pool LSN-POOL-2
```

```

Done

bind lsn pool LSN-POOL-2 203.0.113.5-203.0.113.10

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

bind lsn group LSN-GROUP-2 -poolname LSN-POOL-2

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

Create an LSN configuration with endpoint-independent mapping for HTTP protocol (port 80) and address-port dependent mapping for SSH protocol (port 22). Also, restrict each subscriber to use a maximum of 1000 NAT ports for TCP protocol and 100 NAT ports for UDP protocol. Restrict each subscriber to have a maximum of 2000 concurrent sessions for TCP protocol. Restrict the group to have a maximum of 30000 concurrent sessions for TCP protocol:

```
add lsn client LSN-CLIENT-3

Done

bind lsn client LSN-CLIENT-3 -network 192.0.3.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-3

Done

bind lsn pool LSN-POOL-3 203.0.113.11

Done

add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-3

Done

bind lsn group LSN-GROUP-3 -poolname LSN-POOL-3

```

```
Done

add lsn appsprofile LSN-APPS-HTTPPROFILE-3 TCP -mapping ENDPOINT-
    INDEPENDENT

Done

bind lsn appsprofile LSN-APPS-HTTPPROFILE-3 80

Done

bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-HTTPPROFILE
    -3

Done

add lsn appsprofile LSN-APPS-SSHPROFILE-3 TCP -mapping ADDRESS-PORT-
    DEPENDENT

Done

bind lsn appsprofile LSN-APPS-SSHPROFILE-3 22

Done

bind lsn group LSN-GROUP-3 -applicationprofilename LSN-APPS-SSHPROFILE
    -3

Done

add lsn transportprofile LSN-TRANS-PROFILE-TCP-3 TCP -portquota 1000 -
    sessionquota 2000 -groupSessionLimit 30000

Done

bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-TCP
    -3

Done

add lsn transportprofile LSN-TRANS-PROFILE-UDP-3 UDP -portquota 100

Done

bind lsn group LSN-GROUP-3 -transportprofilename LSN-TRANS-PROFILE-UDP
```

```
    -3

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

Create an LSN configuration for a large set of subscribers:

```
add lsn client LSN-CLIENT-4

Done

bind lsn client LSN-CLIENT-4 -network 192.0.4.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.5.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.6.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.7.0 -netmask 255.255.255.0

Done

bind lsn client LSN-CLIENT-4 -network 192.0.8.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-4

Done

bind lsn pool LSN-POOL-4 203.0.113.30-203.0.113.40

Done

bind lsn pool LSN-POOL-4 203.0.113.45-203.0.113.50

Done

bind lsn pool LSN-POOL-4 203.0.113.55-203.0.113.60

```

```
Done

add lsn group LSN-GROUP-4 -clientname LSN-CLIENT-4

Done

bind lsn group LSN-GROUP-4 -poolname LSN-POOL-4

Done

add lsn appsprofile LSN-APPS-WELLKNOWNPROFILE-4 TCP -mapping ENDPOINT-
   INDEPENDENT

Done

bind lsn appsprofile LSN-APPS-WELLKNOWN-PORTS-PROFILE-4 1- 1023

Done

bind lsn group LSN-GROUP-4 -applicationprofilename LSN-APPS-WELLKNOWN-
   PORTS-PROFILE-4

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

Create an LSN configuration with sharing of NAT resources among multiple LSN groups. In this example, LSN pool LSN-POOL-5 is shared with LSN groups LSN-GROUP-5 and LSN-GROUP-6:

```
add lsn client LSN-CLIENT-5

Done

bind lsn client LSN-CLIENT-5 -network 192.0.15.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-5

Done

bind lsn pool LSN-POOL-5 203.0.113.12-203.0.113.14

Done

add lsn group LSN-GROUP-5 -clientname LSN-CLIENT-5
```

```

Done

bind lsn group LSN-GROUP-5 -poolname LSN-POOL-5

Done

add lsn client LSN-CLIENT-6

Done

bind lsn client LSN-CLIENT-6 -network 192.0.16.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-6

Done

bind lsn pool LSN-POOL-6 203.0.113.15-203.0.113.18

Done

add lsn group LSN-GROUP-6 -clientname LSN-CLIENT-6

Done

bind lsn group LSN-GROUP-6 -poolname LSN-POOL-6

Done

bind lsn group LSN-GROUP-6 -poolname LSN-POOL-5

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

Create an LSN configuration with deterministic NAT resource allocation:

```
add lsn client LSN-CLIENT-7

Done

bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask 255.255.255.0

Done
```

```

add lsn pool LSN-POOL-7 -nattype DETERMINISTIC

Done

bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23

Done

add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
    DETERMINISTIC -portblocksize 1024

Done

bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7

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

Create an LSN configuration with multiple subscriber networks having the same network address but each network belonging to a different traffic domain. Also, restrict the outbound traffic related to HTTP protocol (port 80), sending it through a particular traffic domain (td 5):

```
add lsn client LSN-CLIENT-8

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
     -td 1

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
     -td 2

Done

bind lsn client LSN-CLIENT-8 -network 192.0.18.0 -netmask 255.255.255.0
     -td 3

Done

add lsn pool LSN-POOL-8

Done
```

```

bind lsn pool LSN-POOL-8 203.0.113.80-203.0.113.86

Done

add lsn group LSN-GROUP-8 -clientname LSN-CLIENT-8

Done

bind lsn group LSN-GROUP-8 -poolname LSN-POOL-8

Done

add lsn appsprofile LSN-APPS-HTTP-PROFILE-8 TCP -td 5

Done

bind lsn appsprofile LSN-APPS-HTTP-PROFILE-8 80

Done

bind lsn group LSN-GROUP-8 -applicationprofilename LSN-APPS-HTTP-
    PROFILE-8

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

Create an LSN configuration that restricts the outbound traffic of a specific protocol (TCP), sending it through a particular traffic domain (td 5). With endpoint-independent filtering, receive inbound traffic related to this protocol (TCP) on any traffic domain:

```
add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.9.0 -netmask 255.255.255.0
    -td 1

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.90
```

```

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

add lsn appsprofile LSN-APPS-PROFILE-9 TCP -filtering ENDPOINT-
    INDEPENDENT -td 5

Done

bind lsn group LSN-GROUP-9 -approfile LSN-APPS-PROFILE-9

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

Create an LSN configuration that restricts outbound HTTP (port 80) traffic, sending it through a particular traffic domain (td 10). With address-dependent filtering, receive inbound traffic related to this protocol (HTTP) on the specified traffic domain (td 10):

```
add lsn client LSN-CLIENT-10

Done

bind lsn client LSN-CLIENT-10 -network 192.0.10.0 -netmask
    255.255.255.0 -td 1

Done

add lsn pool LSN-POOL-10

Done

bind lsn pool LSN-POOL-10 203.0.113.100

Done

add lsn group LSN-GROUP-10 -clientname LSN-CLIENT-10

Done
```

```

bind lsn group LSN-GROUP-10 -poolname LSN-POOL-10

Done

add lsn appsprofile LSN-APPS-PROFILE-10 TCP -mapping ENDPOINT -INDEPENDENT -filtering ADDRESS-DEPENDENT -td 10

Done

bind lsn appsprofile LSN-APPS-PROFILE-10 80

Done

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

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

Configuring Static LSN Maps

September 21, 2020

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

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

At the command prompt, type:

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

-  show lsn static
<!--NeedCopy-->
```

To create a static LSN mapping by using the configuration utility

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

Parameter Descriptions (of commands listed in the CLI procedure)

add lsn static name

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

transportprotocol

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

subscrIP

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

subscrPort

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

td

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

natIP

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

natPort

NAT port for this LSN mapping entry.

destIP

Destination IP address for the LSN mapping entry.

dsttd

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

Wildcard Port Static Maps

A static mapping entry is usually a one-to-one LSN mapping between a subscriber IP address:port and a NAT IP address:port. A one-to-one static LSN mapping entry exposes only one port of the subscriber to the Internet.

Some situations might require exposing all ports (64K) of a subscriber to the Internet (for example, a server hosted on an internal network and running a different service on each port). To make these internal services accessible through the Internet, you have to expose all the ports of the server to the Internet.

One way to meet this requirement is to add 64K one-to-one static mapping entries, one mapping entry for each port. Creating 64K entries is very cumbersome and a big task. Also, this large number of configuration entries might lead to performance issues in the NetScaler appliance.

Another simple method is to use wildcard ports in a static mapping entry. You just need to create one static mapping entry with NAT-port and subscriber-port parameters set to the wildcard character (*), and the protocol parameter set to ALL, to expose all the ports of a subscriber to the Internet. For a subscriber's inbound or outbound connections matching a wildcard static mapping entry, the subscriber's port does not change after the NAT operation.

When a subscriber-initiated connection to the Internet matches a wildcard static mapping entry, the NetScaler appliance assigns a NAT port that has the same number as the subscriber port from which the connection is initiated. Similarly, an Internet host gets connected to a subscriber's port by connecting to the NAT port that has the same number as the subscriber's port.

Configuring the NetScaler appliance to Provide Access to All Ports of an IPv4 Subscriber

To configure the NetScaler appliance to provide access to all ports of an IPv4 subscriber, create a wildcard static map with the following mandatory parameter settings:

- Protocol=ALL

- Subscriber port = *
- NAT port = *

In a wildcard static map, unlike in a one-to-one static map, setting the NAT IP parameter is mandatory. Also, the NAT IP address assigned to a wildcard static map cannot be used for any other subscribers.

To create a wildcard static map by using the command line interface

At the command prompt, type:

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <
    positive_integer>] [-destIP <ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

Sample Configuration

In the following sample configuration of a wildcard static map, all ports of a subscriber whose IP address is 192.0.2.10 are made accessible through NAT IP 203.0.11.33.

Sample configuration:

```
add lsn static NAT44-WILDCARD-STATIC-1 ALL  192.0.2.10 * 203.0.113.33 *

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

Configuring Application Layer Gateways

September 21, 2020

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

The NetScaler appliance supports ALG for the following protocols:

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

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 21, 2020

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

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

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

Sample LSN Configuration for FTP ALG:
In the following sample LSN configuration, FTP ALG is enabled for subscribers that have IP address in the range 192.0.2.30-192.0.2.100.

```
add ns acl LSN-ACL-1 ALLOW -srcIP 192.0.2.30-192.0.2.100

Done

apply acls

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1  - aclname LSN-ACL

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.10

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -FTP ENABLED

Done

```

```
bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

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

Sample LSN Configuration for TFTP ALG:

In the following sample LSN configuration, endpoint-independent mapping and endpoint-independent filtering are enabled for TFTP protocol (UDP port 69). The NetScaler appliance automatically enables TFTP ALG for this LSN configuration.

```
add lsn client LSN-CLIENT-2

Done

bind lsn client LSN-CLIENT-2 -network 198.51.100.0 -netmask 255
    .255.255.0

Done

add lsn pool LSN-POOL-2

Done

bind lsn pool LSN-POOL-2 203.0.113.10-203.0.113.11

Done

add lsn group LSN-GROUP-2 -clientname LSN-CLIENT-2

Done

bind lsn group LSN-GROUP-2 -poolname pool1 LSN-POOL-2

Done

add lsn appsprofile LSNAPPSPROFILE-TFTP-2 UDP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-TFTP-2 69

Done

```

```
bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TFTP
    -2

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

Application Layer Gateway for PPTP Protocol

September 21, 2020

The NetScaler appliance supports Application Layer Gateways (ALGs) for the Point-to-Point Tunneling Protocol (PPTP).

PPTP is a network protocol that enables secure transfer of data from a remote client to an enterprise server by creating a tunnel across TCP/IP-based data networks. PPTP encapsulates PPP packets into IP packets for transmission over the Internet. PPTP establishes a tunnel for each communicating PPTP network server (PNS)-PPTP Access Concentrator (PAC) pair. After the tunnel is set up, enhanced generic routing encapsulation (GRE) is used to exchange PPP packets. A call ID in the GRE header indicates the session to which a particular PPP packet belongs.

The NetScaler appliance recognizes PPTP packets that arrive on the default TCP port, 1723. The appliance parses PPTP control packets, translates the call ID, and assigns a NAT IP address. For two-way data communication between the client and server, the NetScaler appliance creates an LSN session entry based on the server call ID, and an LSN session based on the client call ID. The appliance then parses the GRE data packets and translates call IDs on the basis of the two LSN session entries.

For PPTP protocol, the NetScaler appliance also includes timeout setting for any idle PPTP LSN sessions. If a PPTP LSN session is idle for a time that exceeds the timeout setting, the NetScaler appliance removes the session.

Limitations:

The following are the limitations of PPTP ALG on a NetScaler appliance:

- PPTP ALG is not supported for hairpin LSN flow.
- PPTP ALG is not supported to work with any RNAT configuration.
- PPTP ALG is not supported inNetScaler clusters.

Configuring PPTP ALG

Configuring PPTP ALG on the NetScaler appliance consist of the following tasks:

- Create an LSN configuration and enable PPTP ALG on it. In an LSN configuration, the LSN group includes the PPTP ALG setting. For instructions on creating an LSN configuration, see Configuration Steps for LSN.
- (Optional) Set the global timeout for idle PPTP LSN sessions.

To enable PPTP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-pptp ( ENABLED |
    DISABLED )]

show lsn group
<!--NeedCopy-->
```

To set the global timeout for idle PPTP LSN sessions by using the CLI

At the command prompt, type:

```
set appAlgParam -pptpGreIdleTimeout <positive_integer>

show appAlgParam
<!--NeedCopy-->
```

Example:

In the following sample LSN configuration, PPTP ALG is enabled for subscribers in the 192.0.2.0/24 network.

Also idle PPTP LSN session timeout is set to 200 secs.

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3
```

```

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -pptp ENABLED

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

set appAlgParam -pptpGreIdleTimeout 200

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

Application Layer Gateway for SIP Protocol

September 21, 2020

Using Large Scale NAT (LSN) with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media.

SIP ALG adheres to the following RFCs:

- RFC 3261
- RFC 3581
- RFC 4566
- RFC 4475

How SIP ALG Works

How IP address translation is performed depends on the type and direction of the message. A message can be any of the following:

- Inbound request
- Outbound response
- Outbound request
- Inbound response

For an outgoing message, the private IP address and port number of the SIP client are replaced with the NetScaler owned public IP address and port number, called the *LSN pool IP address and port number*, specified during LSN configuration. For an incoming message, the LSN pool IP address and the port number are replaced with the private address of the client. If the message contains any public IP addresses, the NetScaler SIP ALG retains them. Also, a pinhole is created on the:

- LSN pool IP address and port on behalf of the private client, so that the messages that arrive at this IP address and port from the public network are treated as SIP messages.
- Public IP address and port on behalf of the public clients, so that the messages that arrive at this IP address and port from the private network are treated as SIP messages.

When a SIP message is sent out across the network, the SIP Application Layer Gateway (ALG) collects information from the message and translates the IP addresses in the following headers into LSN pool IP addresses:

- Via
- Contact
- Route
- Record-Route

In the following sample SIP request message, LSN replaces the IP addresses in the header fields to hide them from the outside network.

```
INVITE adam@10.102.185.156 SIP/2.0 Via: SIP/2.0/UDP 192.170.1.161:62914
     From: eve@10.120.210.3 To: adam@10.102.185.156 Call-ID: a12abcde@10
    .120.210.3 Contact: adam@10.102.185.156 Route: <sip:netscreen@10
    .150.20.3:5060> Record-Route: <sip:netscreen@10.150.20.3:5060>
<!--NeedCopy-->
```

When a message containing SDP information arrives, the SIP ALG collects information from the message and translates the IP addresses in the following fields into LSN pool IP addresses and port numbers:

- c= (connection information)

  This field can appear at the session or media level. It appears in the following format:

  `c=<network-type><address-type><connection-address>`

  If the destination IP address is a unicast IP address, the SIP ALG creates pinholes by using the IP address and port numbers specified in the m= field.

- m= (media announcement)

  This field appears at the media level and contains the description of the media. It appears in the following format:

  `m=<media><port><transport><fmt list>`

- a= (information about the media field)

  This field can appear at the session or media level, in the following format:

  `a=<attribute>`

  `a=<attribute>:<value>`

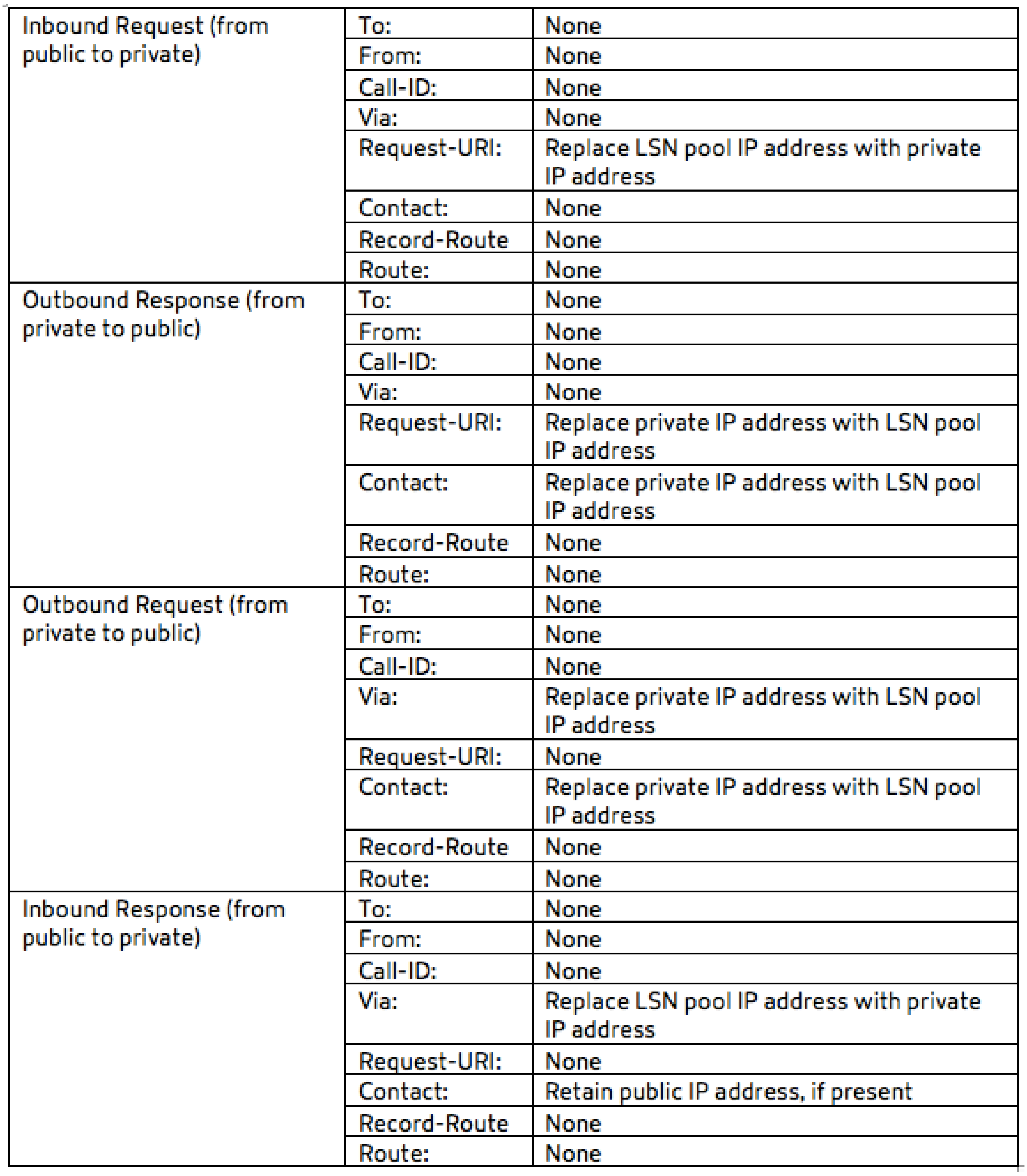
The following excerpt from a sample SDP section shows the fields that are translated for resource allocation.

o=user 2344234 55234434 IN IP4 10.150.20.3

c=IN IP4 10.150.20.3

m=audio 43249 RTP/AVP 0

The following table shows how SIP payload is translated.



Inbound Request (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace LSN pool IP address with private IP address
	Contact:	None
	Record-Route	None
	Route:	None
Outbound Response (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	None
	Request-URI:	Replace private IP address with LSN pool IP address
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Outbound Request (from private to public)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace private IP address with LSN pool IP address
	Request-URI:	None
	Contact:	Replace private IP address with LSN pool IP address
	Record-Route	None
	Route:	None
Inbound Response (from public to private)	To:	None
	From:	None
	Call-ID:	None
	Via:	Replace LSN pool IP address with private IP address
	Request-URI:	None
	Contact:	Retain public IP address, if present
	Record-Route	None
	Route:	None

Limitations of SIP ALG

A SIP ALG has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses

  - Encrypted SDP
  - SIP TLS
  - FQDN translation
  - SIP layer authentication
  - TD/partitioning/cluster
  - Multipart body
  - SIP messages over IPv6 network
  - Line folding

Tested SIP Clients and Proxy Servers

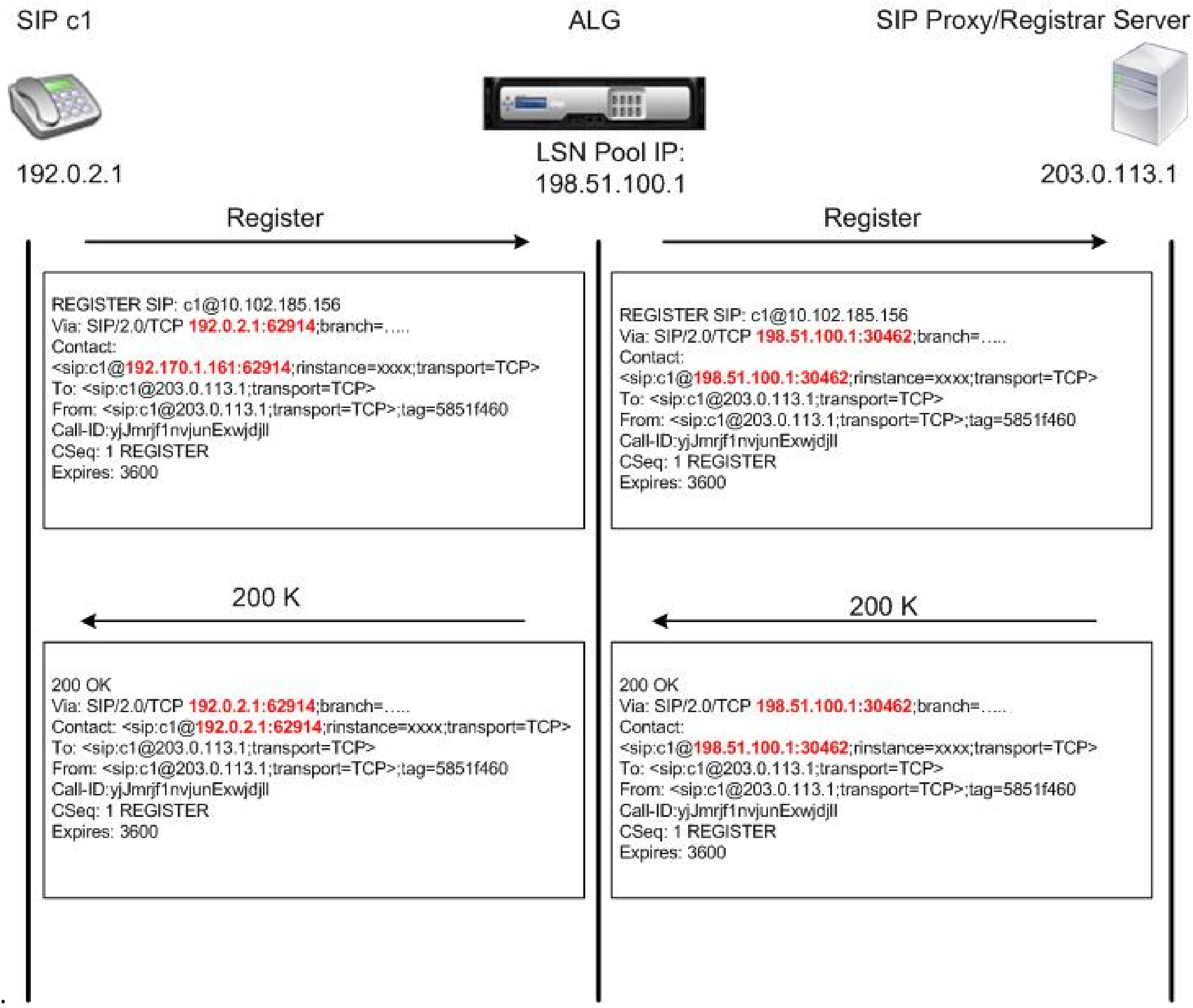
The following SIP clients and proxy server have been tested with SIP ALG:

- **SIP Clients:** X-Lite, Zoiper, Ekiga. Avaya
- **Proxy Server:** openSIPS

LSN SIP Scenario: SIP Proxy Outside the Private Network (Public Network)

SIP Client Registration

For a typical SIP call, SIP client must register with the SIP registrar by composing a REGISTER request and sending it to the SIP registrar. The NetScaler appliance's SIP ALG intercepts the request, replaces the IP address and port number in the request with the LSN pool IP address and port number provided in the LSN configuration, and forwards the request to the SIP registrar. The SIP ALG then opens a pinhole in the NetScaler configuration to allow further SIP communication between the SIP client and the SIP registrar. The SIP registrar sends a 200 OK response to the SIP client over the LSN pool IP address and port number. The NetScaler appliance captures this response in the pinhole, and the SIP ALG replaces the SIP header, putting the original Contact, Via, Route, and Record-Route SIP fields back into the message. The SIP ALG then forwards the message to the SIP client. The following figure shows how SIP ALG uses LSN in a SIP call registration



```
REGISTER SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact:
<sip:c1@192.170.1.161:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

```
REGISTER SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

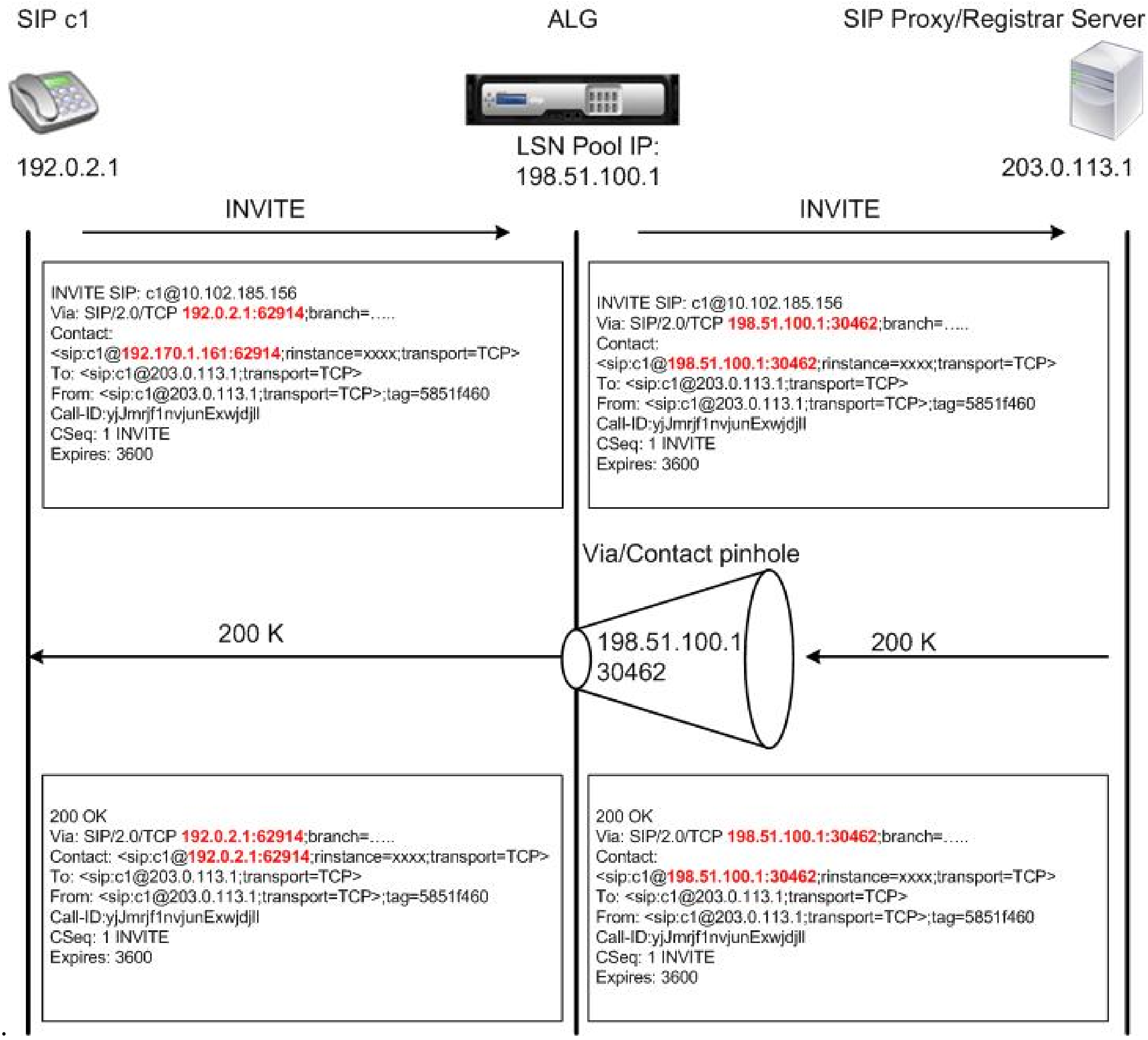
```
200 OK
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact: <sip:c1@192.0.2.1:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

```
200 OK
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

flow.

Outgoing Calls

A SIP call is initiated with a SIP INVITE message sent from the internal to the external network. The SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, replacing them with the LSN pool IP address and port number. LSN stores these mappings for subsequent SIP messages in the SIP call. The SIP ALG then opens separate pinholes in the NetScaler configuration to allow SIP and media through the NetScaler appliance on the dynamically assigned ports specified in the SDP and SIP headers. When a 200 OK message arrives at the NetScaler, it is captured by one of the created pinholes. The SIP ALG replaces the SIP header, restoring the original Contact, Via, Route, and Record-Route SIP fields, and then forwards the message to the internal SIP client.



```
INVITE SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact:
<sip:c1@192.170.1.161:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 INVITE
Expires: 3600
```

```
INVITE SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 INVITE
Expires: 3600
```

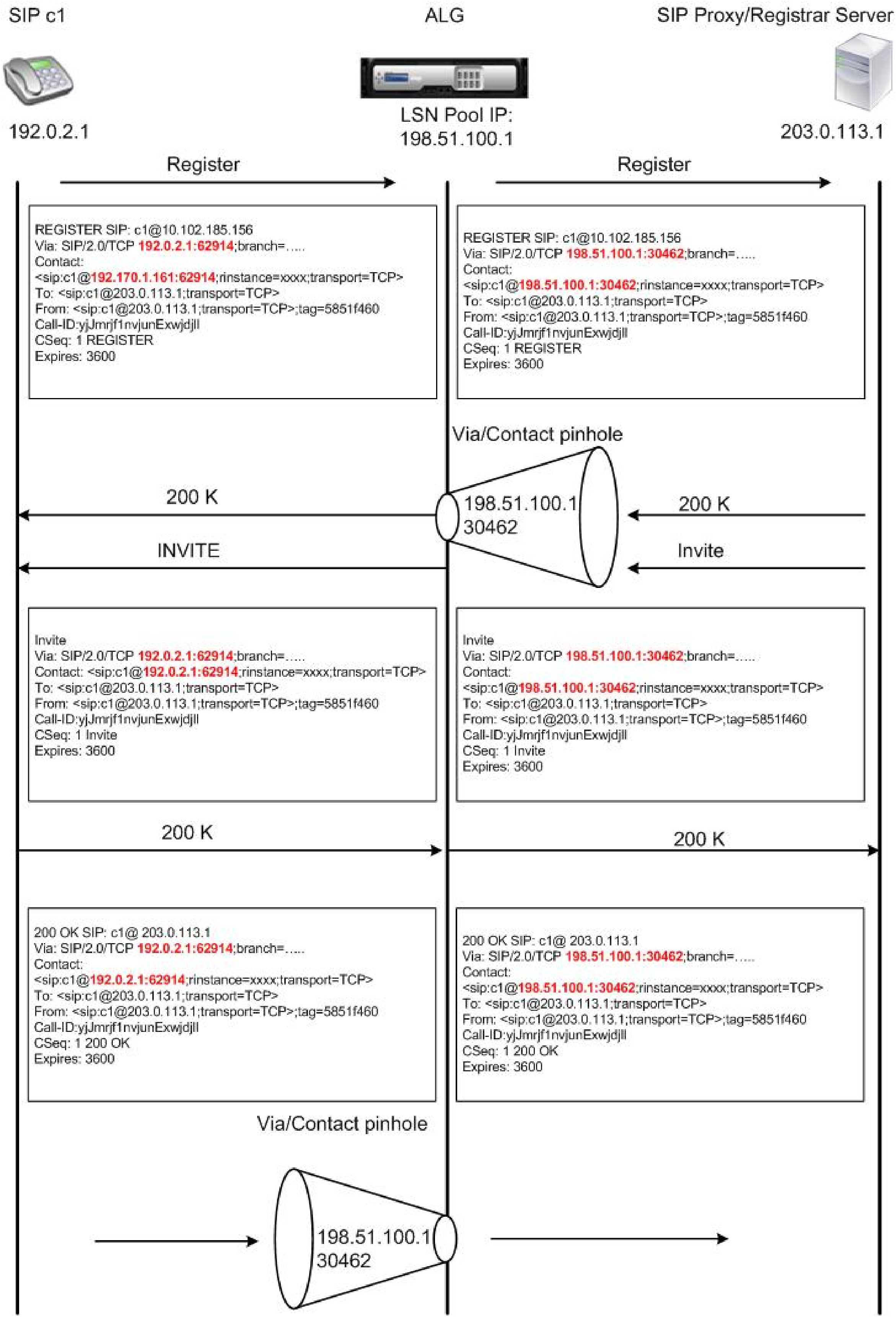
```
200 OK
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact: <sip:c1@192.0.2.1:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 INVITE
Expires: 3600
```

```
200 OK
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 INVITE
Expires: 3600
```

Incoming Calls

A SIP incoming call is initiated with a SIP INVITE message from the external client to the internal network. The SIP registrar forwards the INVITE message to the SIP client in the internal network, using the pinhole that was created when the Internal SIP client registered with the SIP registrar.

The SIP ALG performs NAT on the LSN IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the IP address and port number of the internal SIP client, and forwards the request to the SIP client. When the 200 OK response message sent by the internal SIP client arrives at the NetScaler appliance, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, translating them to the LSN pool IP address and port number, forwards the response message to the SIP registrar, and then opens a pinhole in the outbound direction for further SIP communica-


SIP c1
ALG
SIP Proxy/Registrar Server
192.0.2.1
LSN Pool IP:
198.51.100.1
203.0.113.1
Register
Register
Via/Contact pinhole
200 K
198.51.100.1
30462
200 K
INVITE
Invite
200 K
200 K
Via/Contact pinhole
198.51.100.1
30462

```
REGISTER SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact:
<sip:c1@192.170.1.161:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

```
REGISTER SIP: c1@10.102.185.156
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 REGISTER
Expires: 3600
```

```
Invite
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact: <sip:c1@192.0.2.1:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 Invite
Expires: 3600
```

```
Invite
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 Invite
Expires: 3600
```

```
200 OK SIP: c1@ 203.0.113.1
Via: SIP/2.0/TCP 192.0.2.1:62914;branch=.....
Contact:
<sip:c1@192.0.2.1:62914;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 200 OK
Expires: 3600
```

```
200 OK SIP: c1@ 203.0.113.1
Via: SIP/2.0/TCP 198.51.100.1:30462;branch=.....
Contact:
<sip:c1@198.51.100.1:30462;rinstance=xxxx;transport=TCP>
To: <sip:c1@203.0.113.1;transport=TCP>
From: <sip:c1@203.0.113.1;transport=TCP>;tag=5851f460
Call-ID:yjJmrjf1nvjunExwjdjll
CSeq: 1 200 OK
Expires: 3600
```

tion.

Call Termination

The BYE message terminates a call. When the device receives a BYE message, it translates the header fields in the message just as it does for any other message. But because a BYE message must be acknowledged by the receiver with a 200 OK, the ALG delays call teardown for 15 seconds to allow time for transmission of the 200 OK.

Call Between Clients in the Same Network

When both client A and client B in the same network initiate a call, the SIP messages are routed through the SIP proxy in the outside network. The SIP ALG processes the INVITE from client A as a normal outgoing call. Since client B is in the same network, the SIP proxy sends the INIVITE back to the NetScaler appliance. The SIP ALG examines the INIVITE message, determines that it contains the NAT IP address of client A, and replaces that address with the private IP address of client A before sending the message to client B. Once the call is established between the clients, the NetScaler is not involved in the media transmission between the clients.

More LSN SIP Scenarios: SIP Proxy Inside the Private Network

If you want to host the SIP Proxy server inside the private network, Citrix recommends that you do one of the following:

- Configure a static LSN Mapping for the private SIP proxy. For more information, see Configuring Static LSN Maps. Make sure that the NAT port is the same as the port configured in the SIP ALG profile.
- Configure the SIP Proxy server inside a demilitarized zone (DMZ).

Figure 1. SIP Call Registration

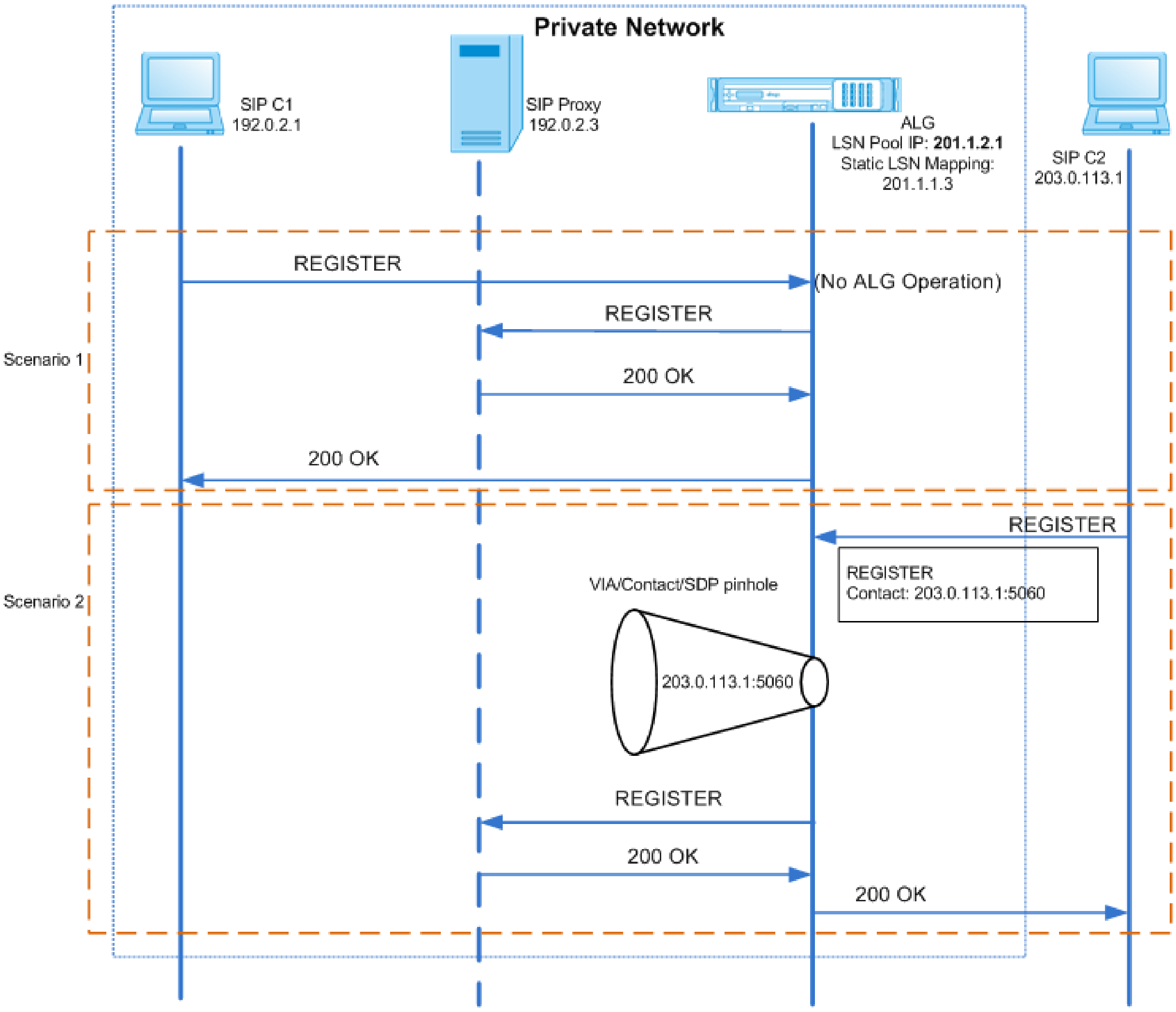
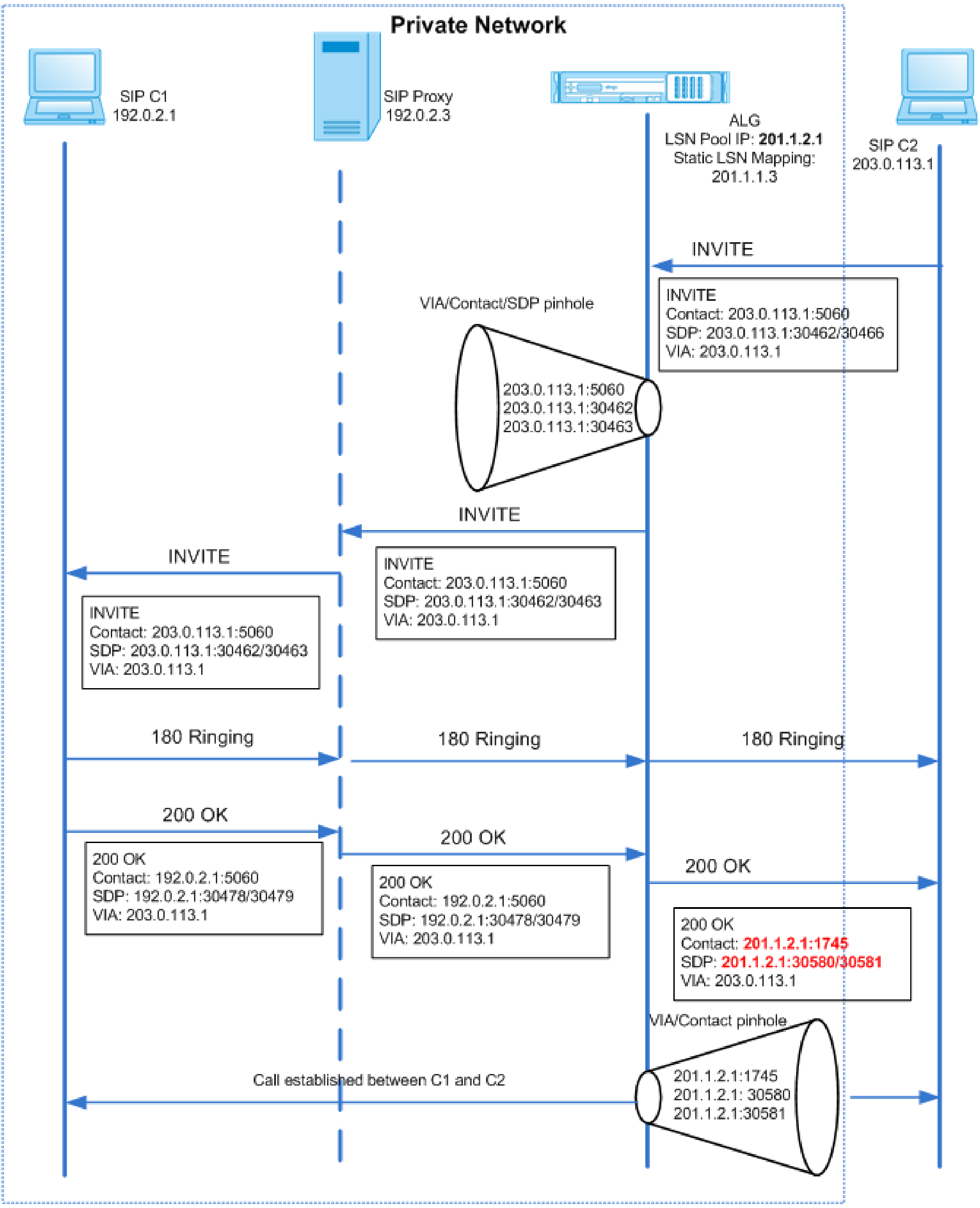

Private Network
SIP C1
192.0.2.1
SIP Proxy
192.0.2.3
ALG
LSN Pool IP: 201.1.2.1
Static LSN Mapping:
201.1.1.3
SIP C2
203.0.113.1
Scenario 1
REGISTER
(No ALG Operation)
REGISTER
200 OK
200 OK
Scenario 2
REGISTER
REGISTER
Contact: 203.0.113.1:5060
VIA/Contact/SDP pinhole
203.0.113.1:5060
REGISTER
200 OK
200 OK

Figure 2. SIP Incoming Call Flow


Private Network
SIP C1
192.0.2.1
SIP Proxy
192.0.2.3
ALG
LSN Pool IP: 201.1.2.1
Static LSN Mapping:
201.1.1.3
SIP C2
203.0.113.1
INVITE
INVITE
Contact: 203.0.113.1:5060
SDP: 203.0.113.1:30462/30466
VIA: 203.0.113.1
VIA/Contact/SDP pinhole
203.0.113.1:5060
203.0.113.1:30462
203.0.113.1:30463
INVITE
INVITE
Contact: 203.0.113.1:5060
SDP: 203.0.113.1:30462/30463
VIA: 203.0.113.1
INVITE
INVITE
Contact: 203.0.113.1:5060
SDP: 203.0.113.1:30462/30463
VIA: 203.0.113.1
180 Ringing
180 Ringing
180 Ringing
200 OK
200 OK
Contact: 192.0.2.1:5060
SDP: 192.0.2.1:30478/30479
VIA: 203.0.113.1
200 OK
200 OK
Contact: 192.0.2.1:5060
SDP: 192.0.2.1:30478/30479
VIA: 203.0.113.1
200 OK
200 OK
Contact: 201.1.2.1:1745
SDP: 201.1.2.1:30580/30581
VIA: 203.0.113.1
VIA/Contact pinhole
201.1.2.1:1745
201.1.2.1: 30580
201.1.2.1:30581
Call established between C1 and C2

Scenario 3

Figures 1 and 2 show the following scenarios:

- Scenario 1—SIP client in the private network registers with the SIP proxy server in the same network. ALG operations are not performed, because the SIP client and SIP proxy server are in the same network.
- Scenario 2—SIP client in the public network registers with the SIP proxy server in the private network. The REGISTER message from the public SIP client is sent to the NetScaler appliance by using the static LSN mapping configured on the appliance, and the appliance creates a pinhole

for further SIP operations.

- Scenario 3— SIP Incoming call flow. A SIP incoming call is initiated with a SIP INVITE message from the external to the internal network. The NetScaler appliance receives the INVITE message from SIP client C2, which is in the external network, through the static LSN maps configured on the NetScaler appliance.
  The appliance creates a pinhole and forwards the INVITE message to the SIP proxy. The SIP proxy then forwards the INVITE message to SIP client C1 in the internal network. SIP client C1 then sends 180 and 200 OK messages to the SIP proxy, which in turn forwards the message to SIP client C2 through the NetScaler appliance.
  When the 200 OK response message sent by internal SIP client C1 arrives at the NetScaler, the SIP ALG performs NAT on the IP addresses and port numbers in the Via, Contact, Route, and Record-Route SIP header fields, and in the SDP fields, replacing them with the LSN pool IP address and port number. The SIP ALG then forwards the response message to SIP client C2 and opens a pinhole in the outbound direction for further SIP communication.

Support for Audit Logs

You can log ALG information as part of LSN logging by enabling ALG in the LSN audit logging configuration. For more information on LSN logging, see Logging and Monitoring LSN. A log message for an ALG entry in the LSN log consists of the following information:

- Time stamp
- Type of SIP message (for example, SIP request)
- Source IP address and port of the SIP client
- Destination IP address and port of the SIP proxy
- NAT IP address and port
- SIP method
- Sequence number
- Whether or not the SIP client is registered
- Caller's user name and domain
- Receiver's user name and domain

Sample audit log:

Request:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 169 0 : Infomsg: "SIP request" - Group: g2
     - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. - Transport
    : TCP - Source_IP: 192.169.1.165 - Source_port: 57952 -
    Destination_IP: 10.102.185.156 - Destination_port: 5060 - Natted_IP:
     10.102.185.191 - Natted_port: 10313 - Method: REGISTER -
    Sequence_Number: 3060 - Register: YES - Content_Type: -
```

```
Caller_user_name: 156_pvt_1 - Callee_user_name: 156_pvt_1 -
    Caller_domain_name: - Callee_domain_name: -
<!--NeedCopy-->
```

Response:

```
07/19/2013:09:49:19 GMT Informational 0-PPE-0 : default ALG
    ALG_SIP_INFO_PACKET_EVENT 170 0 : Infomsg: "SIP response" - Group:
    g2 - Call_ID: NTY0YjYwMTJmYjNhNDU5ZjlhMmQxOTM5ZTE3Zjc3NjM. -
    Transport: TCP - Response_code 200 - Source_IP: 10.102.185.156 -
    Source_port: 5060 - Destination_IP: 192.169.1.165 - Destination_port
    : 57952 - Natted_IP: 10.102.185.191 - Natted_port: 10313 -
    Sequence_Number: 3060 - Content_Type: - Caller_user_name: 156_pvt_1
    - Callee_user_name: 156_pvt_1 - Caller_domain_name: -
    Callee_domain_name: -
<!--NeedCopy-->
```

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT

Important: For the SIP ALG to work, a full cone NAT configuration is mandatory.

Example:

```
add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT
<!--NeedCopy-->
```

- Create a SIP ALG profile and make sure that you define either the source port range or destination port range.

Example:

```
add lsn sipalgprofile sipalgprofile_tcp -sipsrcportrange 1-65535 -
    sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole
     ENABLED  - sipTransportProtocol TCP
<!--NeedCopy-->
```

- Set SIP ALG = ENABLED, while creating the LSN group.

Example:

```
add lsn group g1 -clientname c1 -sipalg ENABLED
<!--NeedCopy-->
```

- Bind the SIP ALG profile to the LSN group.

Sample SIP ALG Configuration:

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, SIP ALG specific setting, and configure SIP ALG:

```
add lsn pool p1

Done

bind lsn pool p1 10.102.185.190

Done

add lsn client c1

Done

bind lsn client c1 -network 192.170.1.0 -netmask 255.255.255.0

Done

add lsn appsprofile app_tcp TCP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app_udp UDP -ippooling PAIRED -mapping ENDPOINT-
    INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app_tcp 1-65535

Done

bind lsn appsprofile app_udp 1-65535

Done
```

```

add lsn sipalgprofile sipalgprofile_tcp -sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -sipTransportProtocol TCP

Done

add lsn sipalgprofile sipalgprofile_udp -sipdstportrange 5060 -openViaPinhole ENABLED -openRecordRoutePinhole ENABLED -sipTransportProtocol UDP

Done

add lsn group g1 -clientname c1 -sipalg ENABLED

Done

bind lsn group g1 -poolname p1

Done

bind lsn group g1 -appsprofilename app_tcp

Done

bind lsn group g1 -appsprofilename app_udp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_tcp

Done

bind lsn group g1 -sipalgprofilename sipalgprofile_udp

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

Application Layer Gateway for RTSP Protocol

September 21, 2020

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. NetScaler functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to NetScaler owned public IP address called an LSN pool IP address.
- Inbound response—LSN pool IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

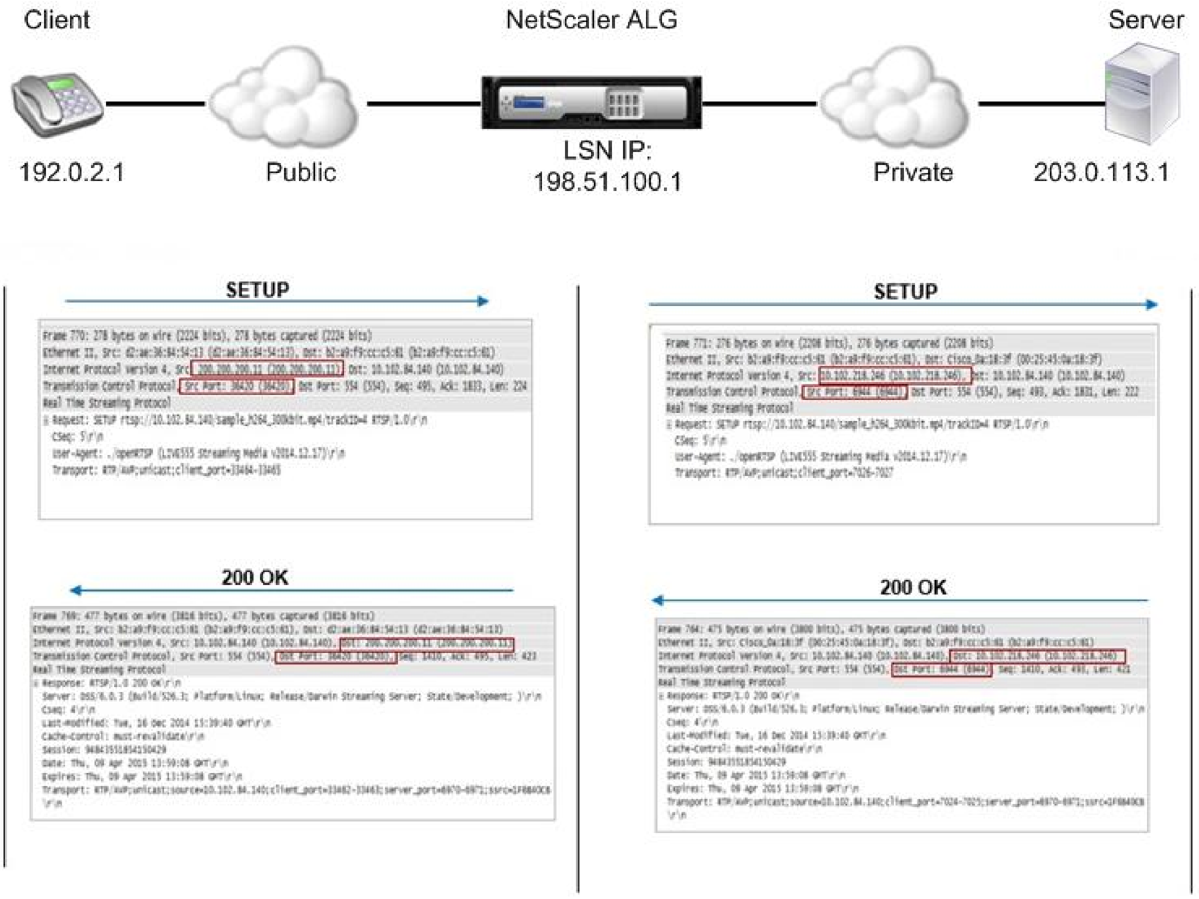
Limitations of RTSP ALG

The RTSP ALG does not support the following:

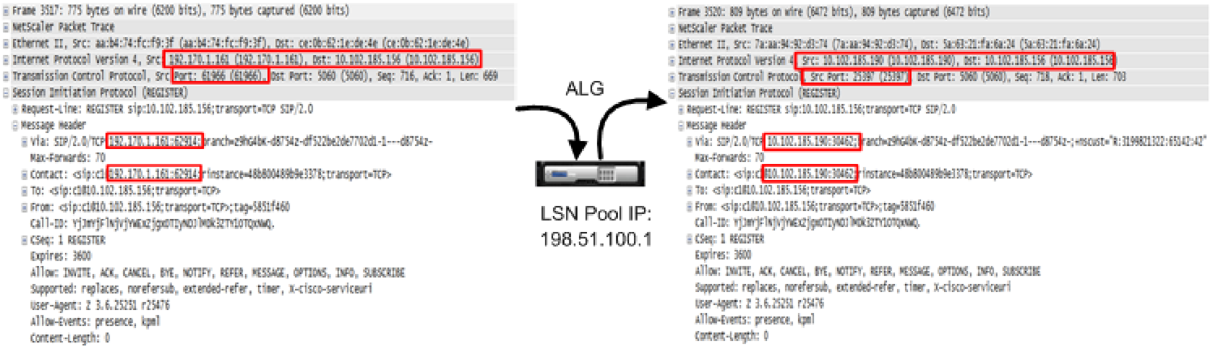
- Multicast RTSP sessions
- RTSP session over UDP
- TD/admin partitioning/cluster deployments
- RSTP Authentication
- HTTP tunneling

RTSP and LSN scenario

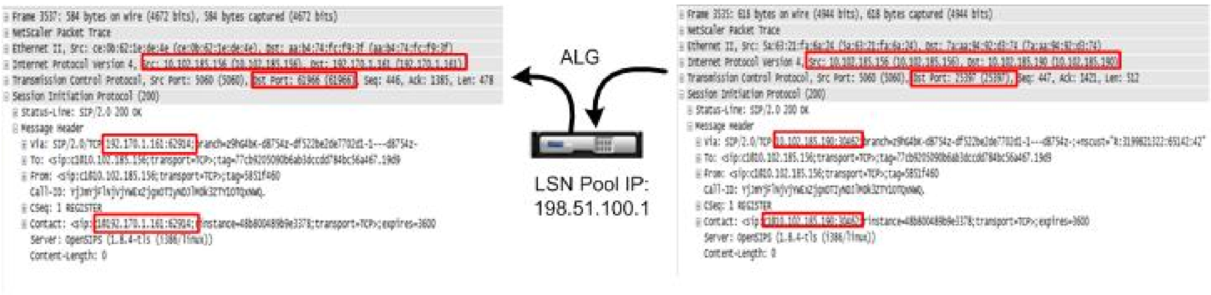
The following figure shows an RTSP SETUP request flow. Typically, a SETUP request specifies how a single media stream must be transported. The request contains the media stream URL and a transport specifier. This specifier typically includes one local port for receiving RTP data (audio or video), and another for receiving RTCP data (meta information). The server reply usually confirms the chosen parameters and fills in the missing parts, such as the server's chosen ports. Each media stream must be configured by using the SETUP command before an aggregate play request can be sent.


Client
NetScaler ALG
Server
192.0.2.1
Public
LSN IP:
198.51.100.1
Private
203.0.113.1
SETUP
200 OK
SETUP
200 OK

In a typical RTSP communication, the media client in the public network sends a SETUP request to the media server in the private network. RSTP ALG intercepts the request and, in the media stream, replaces the public IP address and port number with the LSN pool IP address and LSN port number. The following figure shows the translation performed by a NetScaler appliance in the media stream for an outbound request:


ALG
LSN Pool IP:
198.51.100.1

The media server in the private network uses the LSN pool IP address and LSN port number to send a 200 OK response to the media client in the public network. The NetScaler RTSP ALG intercepts the response and replaces the LSN pool IP address and LSN port number with the public IP address and port number of the media client. The following figure shows the translation performed by a NetScaler appliance in the media stream for an inbound response:



Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Steps for LSN. While configuring LSN, make sure that you:

- Set the **NAT Type** as DETERMINSTIC or DYNAMIC while adding the LSN pool.
- Set the following parameters while adding the LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

Sample RTSP ALG Configuration:

The following sample configuration shows how to create a simple LSN configuration with a single subscriber network, single LSN NAT IP address, and RTSP ALG settings:

```
enable ns feature WL SP LB CS LSN

Done

add lsn pool pool1 -nattype DETERMINISTIC

Done

bind lsn pool pool1 10.102.218.246

Done

add lsn client client1

Done

bind lsn client client1 -network 200.200.200.11 -netmask 255.255.255.0

```

```
Done

add lsn appsprofile app1 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

add lsn appsprofile app2 UDP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile app1 1-65535

Done

bind lsn appsprofile app2 1-65535

Done

add lsn rtspalgprofile rtspalgprofiledefault -rtspIdleTimeout 1000 -rtspportrange 554

Done

add lsn group group1 -clientname client1 -nattype DETERMINISTIC -portblocksize 512 -rtspalg ENABLED

Done

bind lsn group group1 -poolname pool1

Done

bind lsn group group1 -appsprofilename app1

Done

bind lsn group group1 -appsprofilename app2

Done

bind lsn group group1 -rtspalgprofilename rtspalgprofiledefault

Done
```

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

Logging and Monitoring LSN

September 21, 2020

You can log LSN information to diagnose, troubleshoot problems, and to meet legal requirements. You can monitor the performance of the LSN feature by using LSN statistical counters and displaying current LSN sessions.

Logging LSN

Logging LSN information is one of the important functions required by the ISPs to meet legal requirements and for identifying the source of traffic at any given time.

A NetScaler appliance logs LSN mapping entries and the LSN sessions created or deleted for each LSN group. You can control logging of LSN information for an LSN group by using the logging and session logging parameters of the LSN group. These are group level parameters and are disabled by default. The NetScaler appliance logs LSN sessions for an LSN group only when both logging and session logging parameters are enabled.

The following table displays the logging behavior for an LSN group for various settings of logging and session logging parameters.

Logging	Session Logging	Logging Behavior
Enabled	Enabled	Logs LSN mapping entries as well as LSN sessions.
Enabled	Disabled	Logs LSN mapping entries but not LSN sessions.
Disabled	Enabled	Logs neither mapping entries nor LSN sessions.

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

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (MAPPING)

- Whether the LSN mapping entry was created or deleted
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
    - Destination IP address and port are not logged for Endpoint-Independent mapping.
    - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
    - Destination IP address and port are logged for Address-Port-Dependent mapping.

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

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp
- Entry type (SESSION)
- Whether the LSN session is created or removed
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

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

For logging LSN information, Citrix recommends:

- Logging the LSN information on external log servers instead of on the NetScaler appliance. Logging on external servers facilitates optimal performance when the appliance creates large amounts of LSN log entries (in order of millions).

- Using SYSLOG over TCP, or NSLOG. By default SYSLOG uses UDP, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data.

  Note:

    - The SYSLOG generated on NetScaler appliance are dynamically sent to the external log servers.
    - When using SYSLOG over TCP, if the TCP connection is down or the SYSLOG server is busy, then the NetScaler appliances stores the logs in buffer and send the data once the connec-

tion is active.

For more information about configuring logging, see Audit Logging.

Configuring LSN logging consists of the following tasks:

- **Configuring the NetScaler appliance for logging.** This task involves creating and setting various entities and parameters of the NetScaler appliance:
  - **Create a SYSLOG or NSLOG audit logging configuration**. Creating an audit logging configuration involves the following tasks:
    - Create a NSLOG or SYSLOG audit action and enable the LSN parameter. Audit actions specify the IP addresses of log servers.
    - Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy. Audit actions specify the IP addresses of log servers. Optionally, you can set the transport method for log messages that are sent to the external log servers. By default UDP is selected, you can set the transport method as TCP for a reliable transport mechanism.Bind the audit policy to system global.
    - Create a SYSLOG or NSLOG audit policy and bind the audit action to the audit policy.
    - Bind the audit policy to system global.

      Note: For an existing audit logging configuration, just enable the LSN parameter for logging LSN information in the server specified by the audit action.
  - **Enable logging and session logging parameters.** Enable logging and session logging parameters either as you add LSN groups or after you have created the groups. The NetScaler appliance generates log messages related to these LSN groups and sends them to the server of those audit actions that have the LSN parameter enabled.
- **Configuring log servers.** This task involves installing SYSLOG or NSLOG packages on the desired servers. This task also involves specifying the NSIP address of the NetScaler appliance in the configuration file of SYSLOG or NSLOG. Specifying the NSIP address enables the server to identify the log information sent by the NetScaler appliance for storing them in a log file.

For more information about configuring logging, see Audit Logging.

SYSLOG Configuration Using the Command Line Interface

To create a SYSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

```
add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel>... [-transport (TCP)] [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

To create a SYSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

```
add audit syslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

To bind a SYSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

```
bind system global [<policyName> [-priority <positive_integer>]]
<!--NeedCopy-->
```

SYSLOG Configuration Using the Configuration Utility

To configure a SYSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog** and, on the Servers tab, add a new auditing server or edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.
3. (Optional) To enable SYSLOG over TCP, select the **TCP Logging** option.

To configure a SYSLOG server policy for LSN logging by using the configuration utility

Navigate to **Systems** > **Auditing** > **Syslog** and, on the **Policies** tab, add a new policy or edit an existing policy.

To bind a SYSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Syslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

NSLOG Configuration Using the Command Line Interface

To create a NSLOG server action for LSN logging by using the command line interface

At the command prompt, type:

```
add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel
    <logLevel> ... [-lsn ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

To create a NSLOG server policy for LSN logging by using the command line interface

At the command prompt, type:

```
add audit nslogPolicy <name> <rule> <action>
<!--NeedCopy-->
```

To bind a NSLOG server policy to system global for LSN logging by using the command line interface

At the command prompt, type:

```
bind system global [<policyName>]
<!--NeedCopy-->
```

NSLOG Configuration Using the Configuration Utility

To configure a NSLOG server action for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Servers** tab, add a new auditing server or, edit an existing server.
2. To enable LSN logging, select the **Large Scale NAT Logging** option.

To configure a NSLOG server policy for LSN logging by using the configuration utility

Navigate to **Systems** > **Auditing** > **Nslog** and, on the **Policies** tab, add a new policy or edit an existing policy.

To bind a NSLOG server policy to system global for LSN logging by using the configuration utility

1. Navigate to **Systems** > **Auditing** > **Nslog**.
2. On the **Policies** tab, in the **Action** list, click **Global Bindings** to bind the audit global policies.

Example

The following configuration specifies two SYSLOG and two NSLOG servers for storing log entries including LSN logs. LSN Logging is configured for LSN groups LSN-GROUP-2 and LSN-GROUP-3.

The NetScaler appliance generates log messages related to LSN mappings and LSN sessions of these LSN groups, and sends them to the specified log servers.

```
add audit syslogAction SYS-ACTION-1 198.51.101.10 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done
bind system global SYSLOG-POLICY-1
Done

add audit syslogAction SYS-ACTION-2 198.51.101.20 -logLevel ALL -lsn
    ENABLED
Done
add audit syslogPolicy SYSLOG-POLICY-2 ns_true SYS-ACTION-2
Done
bind system global SYSLOG-POLICY-2
Done

add audit nslogAction NSLOG-ACTION-1 198.51.101.30 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-1 ns_true NSLOG-ACTION-1
Done
bind system global NSLOG-POLICY-1
Done
add audit nslogAction NSLOG-ACTION-2 198.51.101.40 -logLevel ALL -lsn
    ENABLED
Done
add audit nslogPolicy NSLOG-POLICY-2 ns_true NSLOG-ACTION-2
Done
bind system global NSLOG-POLICY-2
Done

 add lsn group LSN-GROUP-3 -clientname LSN-CLIENT-2  – logging ENABLED
     – sessionLogging ENABLED
Done
 set lsn group LSN-GROUP-2   – logging ENABLED  – sessionLogging ENABLED
Done
<!--NeedCopy-->
```

The following configuration specifies SYSLOG configuration for sending log messages to the external SYSLOG server 192.0.2.10 using TCP.

```
add audit syslogAction SYS-ACTION-1 192.0.2.10 -logLevel ALL -transport
     TCP
Done
```

```

add audit syslogPolicy SYSLOG-POLICY-1 ns_true SYS-ACTION-1
Done

bind system global SYSLOG-POLICY-1
Done
<!--NeedCopy-->
```

The following table displays sample LSN log entries of each type stored on the configured log servers. These LSN log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
LSN session creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:48 GMT 0-PPE-0 : LSN LSN_SESSION 2581750 : SESSION CREATED Client IP:Port:TD 192.0.2.10: 15136:0, NatIP:NatPort 203.0.113.6: 6234, Destination IP:Port:TD 198.51.100.9: 80:0, Protocol: TCP
LSN session deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_SESSION 3871790 : SESSION DELETED Client IP:Port:TD 192.0.2.11: 15130:0, NatIP:NatPort 203.0.113.6: 7887, Destination IP:Port:TD 198.51.101.2:80:0, Protocol: TCP
LSN mapping creation	Local4.Informational 10.102.37.115 08/05/2014:09:59:47 GMT 0-PPE-0 : LSN LSN_MAPPING 2581580 : EIM CREATED Client IP:Port 192.0.2.15: 14567, NatIP:NatPort 203.0.113.5: 8214, Protocol: TCP
LSN mapping deletion	Local4.Informational 10.102.37.115 08/05/2014:10:05:12 GMT 0-PPE-0 : LSN LSN_MAPPING 3871700 : EIM DELETED Client IP:Port 192.0.3.15: 14565, NatIP:NatPort 203.0.113.11: 8217, Protocol: TCP

Minimal Logging

Deterministic LSN configurations and Dynamic LSN configurations with port block significantly reduces the LSN log volume. For these two types of configuration, the NetScaler appliance allocates a NAT IP address and a block of ports to a subscriber. The NetScaler appliance generates a log message for a port block at the time of allocation to a subscriber. The NetScaler appliance also generates a log message when a NAT IP address and port block is freed. For a connection, a subscriber can be identified just by its mapped NAT IP address and port block. Because of this reason, the NetScaler appliance does not log any LSN session created or deleted. The appliance also neither logs any mapping entry created for a session nor when the mapping entry gets removed.

The minimal logging feature for deterministic LSN configurations and dynamic LSN configurations with port block is enabled by default and there is no provision to disable it. In other words, the NetScaler appliance automatically do minimal logging for deterministic LSN configurations and dynamic LSN configurations with port block. There is no option available for disabling this feature. The appliance sends the log messages to all the configured log servers.

A log message for each port block consists of the following information:

- NSIP address of the NetScaler appliance
- Time stamp
- Entry type as DETERMINISTIC or PORTBLOCK
- Whether a port block is allocated or is freed
- Subscriber's IP address and the assigned NAT IP address and port block
- Protocol name

Minimal Logging for Deterministic LSN Configuration

Consider an example of a simple deterministic LSN configuration for four subscribers having the IP address 192.0.17.1, 192.0.17.2, 192.0.17.3, and 192.0.17.4.

In this LSN configuration, the port block size is set to 32768 and LSN NAT IP address pool has IP addresses in the range 203.0.113.19-203.0.113.23.

```
add lsn client LSN-CLIENT-7
Done
bind lsn client LSN-CLIENT-7 -network 192.0.17.0 -netmask
   255.255.255.253
Done
add lsn pool LSN-POOL-7 -nattype DETERMINISTIC
Done
bind lsn pool LSN-POOL-7 203.0.113.19-203.0.113.23
Done
add lsn group LSN-GROUP-7 -clientname LSN-CLIENT-7 -nattype
   DETERMINISTIC -portblocksize 32768
```

```
Done
bind lsn group LSN-GROUP-7 -poolname LSN-POOL-7
Done
<!--NeedCopy-->
```

The NetScaler appliance sequentially preallocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-33791) on the beginning NAT IP address (203.0.113.19) to the beginning subscriber IP address (192.0.17.1). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The appliance logs the NAT IP address and the block of ports allocated for each subscriber.

The NetScaler appliance does not log any LSN session created or deleted for these subscribers. The appliance generates the following log messages for the LSN configuration.

```
1) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201453 0 :  Dtrstc ALLOC Client 12.0.0.241,
   NatInfo 50.0.0.2:59904 to 60415
2) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201454 0 :  Dtrstc ALLOC Client 12.0.0.242,
   NatInfo 50.0.0.2:60416 to 60927
3) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
   NatInfo 50.0.0.2:60928 to 61439
4) 03/23/2015:00:30:56 GMT  Informational 0-PPE-0 : default LSN
   LSN_DETERMINISTIC 79201455 0 :  Dtrstc ALLOC Client 12.0.0.243,
   NatInfo 50.0.0.2:60928 to 61439
<!--NeedCopy-->
```

When you remove the LSN configuration, the allocated NAT IP address and block of ports is freed from each subscriber. The appliance logs NAT IP address and block of ports freed from each subscriber. The appliance generates the following log messages for each subscriber when you remove the LSN configuration.

“‘ pre codeblock
1) 03/23/2015:00:33:57 GMT Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201706 0 : Dtrstc FREE Client 12.0.0.238, NatInfo 50.0.0.2:58368 to 58879
2) 03/23/2015:00:33:57 GMT Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201707 0 : Dtrstc FREE Client 12.0.0.239, NatInfo 50.0.0.2:58880 to 59391
3) 03/23/2015:00:33:57 GMT Informational 0-PPE-0 : default LSN LSN_DETERMINISTIC 79201708 0 : Dtrstc FREE Client 12.0.0.240, NatInfo 50.0.0.2:59392 to 59903

```
### Minimal Logging for Dynamic LSN Configuration with Port Block
```

```

Consider an example of a simple dynamic LSN configuration with port
    block for any subscriber in the network 192.0.2.0/24. In this LSN
    configuration, the port block size is set to 1024 and LSN NAT IP
    address pool has IP addresses in the range 203.0.113.3-203.0.113.4.
```

set lsn parameter -memLimit 4000

Done

add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

Done

```
The NetScaler appliance allocates a random NAT IP address and a block
    of ports, from the LSN NAT IP pool and on the basis of the set port
    block size, for a subscriber when it initiates a session for the
    first time. The NetScaler logs the NAT IP address and block of ports
     allocated to this subscriber. The appliance does not log any LSN
    session created or deleted for this subscriber. If all the ports are
     allocated (for different subscriber’ s sessions) from the
    subscriber’ s allocated port block, the appliance allocates a new
    random NAT IP address and port block for the subscriber for
    additional sessions. The NetScaler logs every NAT IP address and
    port block allocated to a subscriber.

The appliance generates the following log message when the subscriber,
    having the IP address 192.0.2.1, initiates a session. The log
    message shows that the appliance has allocated NAT IP address
    203.0.113.3 and port block 1024-2047 to the subscriber.
```

03/23/2015:00:07:12 GMT Informational 0-PPE-3 : default LSN LSN_PORTBLOCK 106725793 0 : Portblock ALLOC Client 12.0.2.72, NatInfo 203.0.113.3:1024 to 2047, Proto:TCP

```
Once there are no more sessions left that is using the allocated NAT IP
     address and one of the ports in the allocated port block, the
```

```
allocated NAT IP address and block of ports is freed from the subscriber. The NetScaler logs that the NAT IP address and the block of ports is freed from the subscriber. The appliance generates the following log messages for the subscriber, having the IP address 192.0.2.1 , when no more sessions are left that is using the allocated NAT IP address ( 203.0.113.3 ) and a port from the allocated port block ( 1024-2047 ). The log message shows that the NAT IP address and port block are freed from the subscriber.
```

03/23/2015:00:11:09 GMT Informational 0-PPE-3 : default LSN LSN_PORTBLOCK 106814342 0 : Portblock FREE Client 12.0.3.122, NatInfo 203.0.113.3: 1024 to 2047, Proto:TC

```
## Load Balancing SYSLOG Servers

The NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

### Load balancing of SYSLOG servers using the command line interface

Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.
```

add service (|) <serviceType (SYSLOGTCP | SYSLOGUDP)>

```
Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.
```

add lb vserver <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod]

```
Bing the service to the load balancing virtual server.
```

Bind lb vserver

```
Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.
```

add syslogaction [-lbVserverName] [-logLevel]

```
Add a SYSLOG policy by specifying the rule and action.
```

add syslogpolicy

```
Bind the SYSLOG policy to the system global for the policy to take effect.
```

bind system global

```
### Load balancing of SYSLOG servers using the configuration utility

1.  Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

    Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

1.  Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

    Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDPas protocol.

1.  Bing the service to the load balancing virtual server to the service.

    Bing the service to the load balancing virtual server.

    Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

1.  Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

    Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

1.  Add a SYSLOG policy by specifying the rule and action.

    Navigate to System > Syslog, click Policies and add a SYSLOG policy.

1.  Bind the SYSLOG policy to the system global for the policy to take effect.

    Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy
```

```
        to system global.

**Example:**

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.
```

```
add service service1 192.0.2.10 SYSLOGUDP 514
Done

add service service2 192.0.2.11 SYSLOGUDP 514
Done

add service service3 192.0.2.11 SYSLOGUDP 514
Done

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
Done

bind lb vserver lbvserver1 service1
Done

bind lb vserver lbvserver1 service2
Done

bind lb vserver lbvserver1 service3
Done

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
Done

add syslogpolicy syspol1 ns_true sysaction1
Done

bind system global syspol1
Done
```

```
## Logging HTTP Header Information

The NetScaler appliance can now log request header information of an HTTP connection that is using the LSN fucntionality of the NetScaler. The following header information of an HTTP request packet can be logged:

-  URL that the HTTP request is destined to.
-  HTTP Method specified in the HTTP request.
-  HTTP version used in the HTTP request.
-  IP address of the subscriber that sent the HTTP request.

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set a subscribers. For example, an ISP can use this feature to find out the most popular websites among a set of subscribers.

An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging. The HTTP header log profile is then bound to an LSN group. The NetScaler appliance then logs HTTP header attributes, which are enabled in the bound HTTP header log profile for logging, of any HTTP requests related to the LSN group. The appliance then sends the log messages to the configured log servers.

An HTTP header log profile can be bound to multiple LSN groups but an LSN group can have only one HTTP header log profile.

### To create an HTTP header log profile by using the the command line interface

At the command prompt, type:
```

add lsn httphdrlogprofile [-logURL (ENABLED | DISABLED)] [-logMethod (ENABLED | DISABLED)] [-logVersion (ENABLED | DISABLED)] [-logHost (ENABLED | DISABLED)]

show lsn httphdrlogprofile

```
### To bind an HTTP header log profile to an LSN group by using the the command line interface

At the command prompt, type:
```

bind lsn group -httphdrlogprofilename

show lsn group

```
### Example

In the following example of an LSN configuration, HTTP header log profile HTTP-Header-LOG-1 is bound to LSN group LSN-GROUP-1. The log profile has all the HTTP attributes (URL, HTTP method, HTTP version, and HOST IP address) enabled for logging so that all these attributes are logged for any HTTP requests from subscribers (in the
network 192.0.2.0/24) related to the LSN group.
```

add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done

set lsn parameter -memLimit 4000
Done

add lsn client LSN-CLIENT-1
Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0
Done

add lsn pool LSN-POOL-1
Done

bind lsn pool LSN-POOL-1 203.0.113.3-203.0.113.4
Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1 -portblocksize 1024
Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1
Done

bind lsn group LSN-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG-1
Done

```
The NetScaler generates the following HTTP header log message when one of the subscriber belonging to the LSN configuration example sends an HTTP request.

The log message tells us that a client having the IP address 192.0.2.33 sends an HTTP request to URL example.com using HTTP method GET and HTTP version 1.1.
```

03/19/2015:16:24:04 GMT Informational 0-PPE-1 : default LSN Message 59 0 : "LSN Client IP:TD 10.102.37.118:0 URL: example.com Host: 192.0.2.33 Version: HTTP1.1 Method: GET"

```
## Logging MSISDN Information

A Mobile Station Integrated Subscriber Directory Number (MSISDN) is a telephone number uniquely identifying a subscriber across multiple mobile networks. The MSISDN is associated with a country code and a national destination code identifying the subscriber's operator.
```

```

You can configure a NetScaler appliance to include MSISDNs in LSN log entries for subscribers in mobile networks. The presence of MSISDNs in the LSN logs helps the administrator in faster and accurate back tracing of a mobile subscriber who has violated a policy or law, or whose information is required by lawful interception agencies.

The following sample LSN log entries include MSISDN information for a connection from a mobile subscriber in an LSN configuration. The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.3:45195.

|Log Entry Type|Sample Log Entry|
|-|-|
|LSN session creation|Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP|
|LSN mapping creation|Oct 14 15:37:30 10.102.37.77 10/14/2015:10:08:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP|
|LSN session deletion|Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_SESSION 25012 0 : SESSION CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP|
|LSN mapping|Oct 14 15:40:30 10.102.37.77 10/14/2015:10:11:14 GMT 0-PPE-6 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM CREATED E164:5556543210 Client IP:Port:TD 192.0.2.50:4649:0, NatIP:NatPort 203.0.113.3:45195, Destination IP:Port:TD 23.0.0.1:0:0, Protocol: TCP|

### Perform the following tasks for including MSISDN information in LSN logs

-  **Create an LSN log profile**. An LSN log profile includes the log subscriber ID parameter, which specifies whether to or not to include the MSISDN information in the LSN logs of an LSN configuration. Enable the log subscriber ID parameter when creating the LSN log profile.
-  **Bind the LSN log profile to an LSN group of an LSN configuration
```

```
**. Bind the created LSN log profile to an LSN group of an LSN configuration by setting the log profile name parameter to the created LSN log profile name. For instructions on configuring Large Scale NAT, see [Configuration Steps for LSN](/en-us/netscaler/11-1/netscaler-support-for-telecom-service-providers/lsn-introduction/configuration-steps-lsn.html).

#### To create an LSN log profile by using the CLI

At the command prompt, type:
```

add lsn logprofile <logprofilename -logSubscriberID (ENABLED | DISABLED)

show lsn logprofile

```
#### To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:
```

bind lsn group -logProfileName

show lsn group

```
**Sample Configuration:**

In this example of LSN configuration, the LSN log profile has the log subscriber ID parameter enabled. The profile is bound to LSN group LSN-GROUP-9. MSISDN information is included in the LSN session and LSN mapping logs for connections from mobile subscribers (in the network 192.0.2.0/24).
```

add lsn logprofile LOG-PROFILE-MSISDN-9 -logSubscriberID ENABLED

Done
add lsn client LSN-CLIENT-9

Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done
add lsn pool LSN-POOL-9

Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9

Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done
bind lsn group LSN-GROUP-9 -logprofilename LOG-PROFILE-MSISDN-9

Done

```
## Displaying Current LSN Sessions

You can display the current LSN sessions for detecting any unwanted or
    inefficient LSN sessions on the NetScaler appliance. You can display
     all or some LSN sessions on the basis of selection parameters.

Note: When more than a million LSN sessions exist on the NetScaler
    appliance, Citrix recommends displaying selected LSN sessions
    instead of all by using the selection parameters.

### Configuration Using the Command Line Interface

#### To display all LSN sessions by using the command line interface

At the command prompt, type:
```

show lsn session

```
#### To display selective LSN sessions by using the command line
    interface

At the command prompt, type:
```

show lsn session [-clientname] [-network [-netmask] [-td]] [-natIP [-natPort]]

```
##### Example

To display all LSN sessions existing on a NetScaler

![lsn-session](/en-us/netscaler/media/show-lsn-session1.png)

To display all LSN sessions related to an LSN client entity LSN-CLIENT
    -2

![lsn-session2](/en-us/netscaler/media/lsn-session-2.png)
Done

To display all LSN sessions that uses 203.0.113.5 as the NAT IP address
```

```

![lsn-session3](/en-us/netscaler/media/lsn-session-3.png)

### Configuration Using the Configuration Utility

#### To display all or selected LSN sessions by using the configuration utility

1.  Navigate to System  > Large Scale NAT > Sessions, and click the NAT44 tab.
1.  For displaying LSN sessions on the basis of selection parameters, click Search.

### Parameter Descriptions (of commands listed in the CLI procedure)

-
    show lsn session

    -
        clientname

        Name of the LSN Client entity. Maximum Length: 127

    -
        network

        IP address or network address of subscriber(s).

    -
        netmask

        Subnet mask for the IP address specified by the network parameter.

        Default value: 255.255.255.255

    -
        td

        Traffic domain ID of the LSN client entity.

        Default value: 0

        Minimum value: 0

        Maximum value: 4094

    -
        natIP

        Mapped NAT IP address used in LSN sessions.

## Displaying LSN Statistics

You can display statistics related to the LSN feature for evaluating
    the performance of the LSN feature or to troubleshoot problems. You
    can display a summary of statistics of the LSN feature or of a
    particular LSN group. The statistical counters reflect events since
    the NetScaler appliance was last restarted. All these counters are
    reset to 0 when the NetScaler appliance is restarted.

### To display all LSN statistics by using the command line interface

At the command prompt, type:
```

stat lsn

```
### To display statistics for a specified LSN group by using the
    command line interface

At the command prompt, type:
```

stat lsn group []

```
#### Example
```

> stat lsn

Large Scale NAT statistics
Rate(/s) Total
LSN TCP Received Packets 0 40
LSN TCP Received Bytes 0 3026
LSN TCP Transmitted Packets 0 40
LSN TCP Transmitted Bytes 0 3026
LSN TCP Dropped Packets 0 0
LSN TCP Current Sessions 0 0
LSN UDP Received Packets 0 0
LSN UDP Received Bytes 0 0
LSN UDP Transmitted Packets 0 0

LSN UDP Transmitted Bytes 0 0
LSN UDP Dropped Packets 0 0
LSN UDP Current Sessions 0 0
LSN ICMP Received Packets 0 982
LSN ICMP Received Bytes 0 96236
LSN ICMP Transmitted Packets 0 0
LSN ICMP Transmitted Bytes 0 0
LSN ICMP Dropped Packets 0 982
LSN ICMP Current Sessions 0 0
LSN Subscribers 0 1

Done

> stat lsn group LSN-GROUP-1

LSN Group Statistics
Rate (/s) Total
TCP Translated Pkts 0 40
TCP Translated Bytes 0 3026
TCP Dropped Pkts 0 0
TCP Current Sessions 0 0
UDP Translated Pkts 0 0
UDP Translated Bytes 0 0
UDP Dropped Pkts 0 0
UDP Current Sessions 0 0
ICMP Translated Pkts 0 0
ICMP Translated Bytes 0 0
ICMP Dropped Pkts 0 0
ICMP Current Sessions 0 0
Current Subscribers 0 1

Done

```
### Parameter Descriptions (of commands listed in the CLI procedure)

-
    stat lsn group

    -
        groupname

        Name of the LSN Group. Maximum Length: 127

    -
```

```
        detail

        Specifies detailed output (including more statistics). The output can be quite voluminous. Without this argument, the output will show only a summary.

    -
        fullValues

        Specifies that numbers and strings should be displayed in their full form. Without this option, long strings are shortened and large numbers are abbreviated.

    -
        ntimes

        The number of times, in intervals of seven seconds, the statistics should be displayed.

        Default value: 1

    -
        logFile

        The name of the log file to be used as input.

    -
        clearstats

        Clear the statsistics / counters

        Possible values: basic, full

## Compact Logging

Logging LSN information is one of the important functions needed by ISPs to meet legal requirements and be able to identify the source of traffic at any given time. This eventually results in a huge volume of log data, requiring the ISPs to make large investments to maintain the logging infrastructure.

Compact logging is a technique for reducing the log size by using a notational change involving short codes for event and protocol names. For example, C for client, SC for session created, and T for TCP. Compact logging results in an average of 40 percent reduction in
```

```
    log size.

The following examples of NAT44 mapping creation log entries show the advantage of compact logging.

|-|-|

|Default logging format|02/02/2016:01:13:01 GMT Informational 0-PPE-2 : default LSN LSN_ADDRPORT_MAPPING 85 0 : A&PDM CREATED ClientIP:Port:TD1.1.1.1:6500:0,NatIP:NatPort8.8.8.8:47902, DestinationIP:Port:TD2.2.2.2:80:0, Protocol: TCP|
|Compact logging format|02/02/2016:01:14:57 GMT Info 0-PE2:default LSN 87 0:A&PDMC|C-1.1.1.1:6500:0|N-8.8.8.9:51066|D-2.2.2.2:80:0|T|

## Configuration Steps

Perform the following tasks for logging LSN information in compact format:

-  **Create an LSN log profile.** An LSN log profile includes the Log Compact parameter, which specifies whether to or not to log information in compact format for an LSN configuration.
-  **Bind the LSN log profile to an LSN group of an LSN configuration.** Bind the created LSN log profile to an LSN group of an LSN configuration by setting the Log Profile Name parameter to the created LSN log profile name. All sessions and mappings for this LSN group are logged in compact format.

### To create an LSN log profile by using the CLI

At the command prompt, type:
```

add lsn logprofile -logCompact (ENABLED|DISABLED)

show lsn logprofile

```
### To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:
```

bind lsn group -logProfileName

show lsn group

```
**Sample configuration:**
```

```
add lsn logprofile  LOG-PROFILE-COMPACT-9 -logCompact ENABLED

Done
add lsn client LSN-CLIENT-9
Done
bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0
Done
add lsn pool LSN-POOL-9
Done
bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4
Done
add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
Done
bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9
Done
bind lsn group  LSN-GROUP-9 –logProfileName LOG-PROFILE-COMPACT-9
Done
```

```
## IPFIX Logging

The NetScaler appliance supports sending information about LSN events in Internet Protocol Flow Information Export (IPFIX) format to the configured set of IPFIX collector(s). The appliance uses the existing AppFlow feature to send LSN events in IPFIX format to the IPFIX collectors.

IPFIX based logging is available for the following large scale NAT44 related events:

-  Creation or deletion of an LSN session.
-  Creation or deletion of an LSN mapping entry.
-  Allocation or de-allocation of port blocks in the context of deterministic NAT.
-  Allocation or de-allocation of port blocks in the context of dynamic NAT.
-  Whenever subscriber session quota is exceeded.

### Points to Consider before you Configure IPFIX logging

Before you start configuring IPSec ALG, consider the following points:

-  You must configure the AppFlow feature and IPFIX collector(s) on the NetScaler appliance. For instructions, see Configuring the AppFlow feature topic.
```

```

### Configuration Steps

Perform the following tasks for logging LSN information in IPFIX format
   :

-  **Enable LSN logging in the AppFlow configuration**. Enable the LSN
   logging parameter as part of AppFlow configuration.
-  **Create an LSN log profile**. An LSN log profile includes the IPFIX
    parameter that enables or disables the log information in IPFIX
   format.
-  **Bind the LSN log profile to an LSN group of an LSN configuration
   **. Bind the LSN log profile to one or multiple LSN group(s). Events
    related to the bound LSN group will be logged in IPFIX format.

#### To enable LSN logging in the AppFlow configuration by using the
   CLI

At the command prompt, type:
```

set appflow param -lsnLogging (ENABLED | DISABLED)

show appflow param

```
#### To create an LSN log profile by using the CLIAt the command prompt

At the command prompt, type:
```

set lsn logprofile -logipfix (ENABLED | DISABLED)

show lsn logprofile

```
#### To bind the LSN log profile to an LSN group of an LSN
   configuration by using the CLI

At the command prompt, type:
```

bind lsn group -logProfileName

show lsn group
“‘

To create an LSN log profile by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Log** tab, and then add a log profile.

To bind the LSN log profile to an LSN group of an LSN configuration by using the GUI

1. Navigate to **System** > **Large Scale NAT** > **LSN Group,** open the **LSN** group.
2. In **Advanced Settings,** click **+ Log Profile** to bind the created Log profile to the LSN group.

STUN Timeout

September 21, 2020

STUN (Session Traversal Utilities for NAT) enables an end host operating behind a NAT device to discover its NAT IP address and NAT port allocated by the NAT device. Interactive communication applications (for example real-time voice, video, and messaging) running on these hosts use the STUN protocol for discovering NAT IP address and port information. This information is used by these applications to connect to their peer applications in the Internet. STUN protocol includes servers, known as STUN servers, residing in the Internet. Using the STUN protocol, an application of an end host sends a request to a known STUN server, which in turn then embeds the NAT IP address and port in the payload of its response packet.

In an LSN deployment of a NetScaler appliance for an ISP, interactive communication applications (for example real-time voice, video, and messaging) running on a subscribers can use the STUN protocol to discover whether it is behind a NAT (NetScaler appliance) device or not. These applications send a request to a known STUN server. On receiving the request, the NetScaler allocates a NAT IP address and a port for this request, creates an LSN session and an LSN mapping entry, translates the packet with the allocated NAT IP address and port, and then forwards the packet to the STUN server. The STUN server embeds the allocated NAT IP address and port in the payload of its response packet. When the subscriber finally receives the packet, from the payload of the packet it learns that it is behind a NAT device, and the NAT IP address and port allocated for the session.

The application then notifies the peer applications that it is reachable at the NAT IP address and the port of the LSN mapping entry created for the STUN session. It notifies by embedding the NAT IP address and port in the payload of the packets sent to the peer applications. For making the application reachable at the same LSN mapping entry for any external application, full Cone NAT (endpoint Independent mapping and Endpoint Independent filtering) is enabled for the LSN configuration on the NetScaler appliance.

The NetScaler detects an LSN session of type STUN if the request packets are destined to TCP or UDP port 3478, and then marks the created mapping entry of type STUN. The NetScaler applies a timeout called, STUN timeout, to the created STUN LSN mapping entry. A STUN timeout is the maximum time that the NetScaler maintains an idle STUN LSN mapping entry since it was last used by any LSN session. If the STUN LSN mapping session is unused for a time that exceeds the STUN timeout, the NetScaler removes the mapping entry.

For an application on a subscriber that use STUN LSN mapping entry to stay available to other peer applications on the Internet, the application periodically sends keep-alive messages to the NetScaler appliance so that the STUN LSN mapping entry does not timeout. A higher frequency of keep-alive messages can have an affect on the performance a subscriber, especially, if the subscriber is a mobile device. A higher value of STUN timeout reduces the frequency of keep-alive messages from a subscriber.

ALGs on the NetScaler appliance do not apply to an LSN session that use a STUN LSN mapping entry because NAT IP address and NAT port are communicated in payload of the packets related to the session.

For subscribers' applications that use STUN protocol, the LSN configuration must have the following settings:

- STUN timeout. In an LSN configuration, the LSN group includes the STUN timeout setting.
- Endpoint-independent mapping and endpoint-independent filtering for STUN protocol ports.

For instructions on creating an LSN configuration, see Configuration Steps for LSN.

Example:

The following sample LSN configuration applies to applications that use STUN protocol over TCP or UDP. STUN timeout is set to 10 mins. Endpoint-independent mapping and endpoint-independent filtering is set for the STUN TCP port (3748) and the STUN UDP port (3748).

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  -stuntimeout 10

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

```

```
Done

add lsn appsprofile LSNAPPSPROFILE-TCP-STUN-1 TCP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-TCP-STUN-1 3748

Done

bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-TCP-
    STUN-1

Done

add lsn appsprofile LSNAPPSPROFILE-UDP-STUN-1 UDP -mapping ENDPOINT-
    INDEPENDENT  - filtering  ENDPOINT-INDEPENDENT

Done

bind lsn appsprofile LSNAPPSPROFILE-UDP-STUN-1 3748

Done

bind lsn group LSN-GROUP-1 -applicationprofilename LSNAPPSPROFILE-UDP-
    STUN-1

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

TCP SYN Idle Timeout

September 21, 2020

SYN idle timeout is the timeout for establishing TCP connections that use LSN on the NetScaler appliance. If a TCP session is not established within the configured timeout period, the NetScaler removes the session. SYN idle timeout is useful in providing protection against SYN flood attacks. In an LSN configuration, the LSN group entity includes the SYN idle timeout setting.

Example:

In the following sample LSN configuration, SYN idle timeout is set to 30 secs for TCP connections related to subscribers from the 192.0.2.0/24 network.

```
add lsn client LSN-CLIENT-1

Done

bind lsn client LSN-CLIENT-1 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-1

Done

bind lsn pool LSN-POOL-1 203.0.113.3

Done

add lsn group LSN-GROUP-1 -clientname LSN-CLIENT-1  - synidletimeout 30

Done

bind lsn group LSN-GROUP-1 -poolname pool1 LSN-POOL-1

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

Overriding LSN configuration with Load Balancing Configuration

September 21, 2020

An LSN configuration takes precedence over any load balancing configuration by default. For overriding the large scale networking (LSN) configuration with the load balancing configuration for traffic matching both configurations, create a net profile with Override LSN parameter enabled and bind this profile to the virtual server of the load balancing configuration. USNIP or USIP settings of the load balancing configuration are applied to the traffic, instead of applying the LSN IP address of the LSN configuration.

This option is useful in an LSN deployment that includes NetScaler appliances and value added services, such as firewall and optimization devices. In this type of deployment, the ingress traffic on the NetScaler appliance is required to pass through these value-added services before an LSN configuration on the appliance is applied to the traffic. For the NetScaler appliance to send the ingress traffic to a value added service, a load balancing configuration is created and override LSN is enabled on the appliance. The load balancing configuration includes value added services, represented as load balancing services, bound to a virtual server of type ANY. The virtual server is configured with listen policies for identifying the traffic to be sent to the value added service.

To enable override lsn in a net profile by using the CLI

To enable override lsn while adding a net profile, at the command prompt, type

```
add netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable override lsn while adding a net profile, at the command prompt, type

```
set netProfile <name> -overrideLsn ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable override lsn in a net profile by using GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Override LSN** parameter while adding or modifying net profiles.

In the following sample configuration, net profile NETPROFILE-OVERRIDELSN-1 has override LSN option enabled and is bound to load balancing virtual server LBVS-1.

Sample configuration:

```
add netprofile NETPROFILE-OVERRIDELSN-1 -overrideLsn ENABLED

Done

set lb vserver LBVS-1 -netprofile NETPROFILE-OVERRIDELSN-1

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

Clearing LSN Sessions

September 21, 2020

You can remove any unwanted or inefficient LSN sessions from the NetScaler appliance. The appliance immediately releases resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected LSN sessions from the NetScaler appliance.

To clear all LSN sessions by using the command line interface

At the command prompt, type:

```
flush lsn session

show lsn session
<!--NeedCopy-->
```

To clear selective LSN sessions by using the command line interface

At the command prompt, type:

```
flush lsn session [-clientname <string>] [-network <ip_addr> [-netmask
    <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <
    port>]]

show lsn session
<!--NeedCopy-->
```

Example

Clear all LSN sessions existing on a NetScaler

```
flush lsn session

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

Clear all LSN sessions related to LSN client entity LSN-CLIENT-1

```
flush lsn session -clientname LSN-CLIENT-1

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

Clear all LSN sessions related to a subscriber network (192.0.2.0) of LSN client entity LSN-CLIENT-2 belonging to traffic domain 100

```
flush lsn session -clientname LSN-CLIENT-2  - network 192.0.2.0  -
    netmask 255.255.255.0  - td 100

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

To clear all LSN sessions by using the configuration utility

Navigate to System > Large Scale NAT > Sessions, and click Flush Sessions.

Parameter Descriptions (of commands listed in the CLI procedure)

- flush lsn session
    - clientname

      Name of the LSN Client entity. Maximum Length: 127
    - network

      IP address or network address of subscriber(s).
    - netmask

      Subnet mask for the IP address specified by the network parameter.

      Default value: 255.255.255.255
    - td

      Traffic domain ID of the LSN client entity.

      Default value: 0

      Minimum value: 0

      Maximum value: 4094
    - natIP

      Mapped NAT IP address used in LSN sessions.

- natPort

  Mapped NAT port used in the LSN sessions.

Load Balancing SYSLOG Servers

September 21, 2020

The NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

```
add service <name>(<IP> | <serverName>) <serviceType (SYSLOGTCP |
    SYSLOGUDP)> <port>
<!--NeedCopy-->
```

Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

```
add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
    <AUDITLOGHASH>]
<!--NeedCopy-->
```

Bind the service to the load balancing virtual server.

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

1. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

```
add syslogaction <name> <serverIP> [-lbVserverName <string>] [-
    logLevel <logLevel>]
<!--NeedCopy-->
```

Add a SYSLOG policy by specifying the rule and action.

```
add syslogpolicy <name> <rule> <action>
<!--NeedCopy-->
```

Bind the SYSLOG policy to the system global for the policy to take effect.

```
bind system global <policyName>
<!--NeedCopy-->
```

Load balancing of SYSLOG servers using the configuration utility

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   Navigate to Traffic Management > Services, click Add and select SYLOGTCP or SYSLOGUDP as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

   Navigate to Traffic Management > Virtual Servers, click Add and select SYLOGTCP or SYSLOGUDPas protocol.

3. Bing the service to the load balancing virtual server to the service.

   Bing the service to the load balancing virtual server.

   Navigate to Traffic Management > Virtual Servers, select a virtual server and then selectAUDITLOGHASH in the Load Balancing Method.

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   Navigate to System > Auditing, click Servers and add a server by selecting LB Vserver option inServers.

5. Add a SYSLOG policy by specifying the rule and action.

   Navigate to System > Syslog, click Policies and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   Navigate to System > Syslog, select a SYSLOG policy and click Action, and then click Global Bindings and bind the policy to system global.

Example:

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514

add service service2 192.0.2.11 SYSLOGUDP 514

add service service3 192.0.2.11 SYSLOGUDP 514

add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH

bind lb vserver lbvserver1 service1

bind lb vserver lbvserver1 service2

bind lb vserver lbvserver1 service3

add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All

add syslogpolicy syspol1 ns_true sysaction1

bind system global syspol1
<!--NeedCopy-->
```

Limitations:

The NetScaler appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

Port Control Protocol

September 21, 2020

NetScaler appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A NetScaler appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- (Optional) Create a PCP profile. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:
  - Mapping: Enabled
  - Peer: Enabled
  - Minimum map life: 120 seconds
  - Maximum max life: 86400 seconds
  - Announce count: 10
  - Third Party: Disabled
- Create a PCP server and bind a PCP profile to it. Create a PCP server on the NetScaler appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- Bind the PCP server to an LSN group of an LSN configuration. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group.

> **Note**
>
> A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer ( ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-announceMultiCount <positive_integer>][-thirdParty ( ENABLED | DISABLED )]
```

```

show pcp profile <name>
<!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

Sample Configuration for NAT44

In the following sample configuration, PCP server PCP-SERVER-9, with default PCP settings, is bound to LSN group LSN-GROUP-9. PCP-SERVER-9 serves PCP requests from subscribers in network 192.0.2.0/24.

Sample configuration:

```
add pcp server PCP-SERVER-9 192.0.3.9

Done

add lsn client LSN-CLIENT-9

Done

bind lsn client LSN-CLIENT-9 -network 192.0.2.0 -netmask 255.255.255.0

Done

add lsn pool LSN-POOL-9

Done

bind lsn pool LSN-POOL-9 203.0.113.3-203.0.113.4

Done

add lsn group LSN-GROUP-9 -clientname LSN-CLIENT-9
```

```

Done

bind lsn group LSN-GROUP-9 -poolname LSN-POOL-9

Done

bind lsn group  LSN-GROUP-9  -pcpServer PCP-SERVER-9

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

Dual-Stack Lite

September 21, 2020

Because of the shortage of IPv4 addresses, and the advantages of IPv6 over IPv4, many ISPs have started transitioning to IPv6 infrastructure. But during the transition, ISPs must continue to support IPv4 along with IPv6, because most of the public Internet still uses only IPv4, and many subscribers do not support IPv6.

Dual Stack Lite (DS-Lite) is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv4 subscribers to the Internet. DS-Lite uses IPv4-in-IPv6 tunneling to send a subscriber's IPv4 packet through a tunnel on the IPv6 access network to the ISP. The IPv6 packet is decapsulated to recover the subscriber's IPv4 packet and is then sent to the Internet after NAT address and port translation and other LSN related processing. The response packets traverse through the same path to the subscriber.

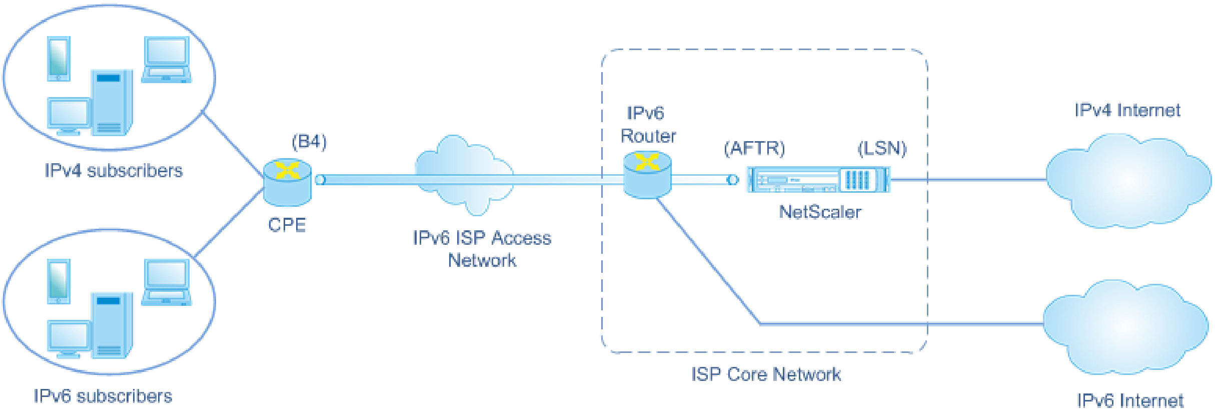
The NetScaler appliance implements the AFTR component of a DS-Lite deployment and is compliant with RFC 6333.

Architecture

The Dual-Stack Lite architecture for an ISP consists of the following components:

- **Basic Bridging Broadband (B4).** Basic Bridging broadband, or B4, is a device or component that resides in the subscriber premises. Typically, B4 is a component in the CPE devices in the subscriber premises. IPv4 subscribers are connected to the IPv6-only ISP access network through the CPE device containing the B4 component. The main function of the B4 is to initiate an IPv6 tunnel between B4 and an address family transition router (AFTR) in order to send or receive subscriber IPv4 request or response packets over the tunnel. B4 includes an IPv6 address known as the B4 tunnel endpoint address. B4 uses this address to source IPv6 packets to AFTR and receive packets from AFTR.

- **Address family transition router (AFTR).** AFTR is a device or component residing in the ISP's core network. AFTR terminates the IPv6 tunnel from the B4 device. In other words, the IPv6 tunnel is formed between B4 in the subscriber premise and AFTR in ISP core network. AFTR decapsulates IPv6 packets received from B4 to recover the subscribers' original IPv4 packets. AFTR sends the IPv4 packets to the LSN device or component. LSN routes the IPv4 packets to their destination after performing NAT address and port translation (NAT 44) and other LSN related processing. AFTR includes an IPv6 address known as the AFTR tunnel endpoint address. AFTR uses this address to source IPv6 packets to B4 and receive IPv6 packets from B4. The NetScaler appliance implements the AFTR component.
- **Softwire.** The IPv6 tunnel created between B4 and AFTR is called a softwire.



The DS-Lite architecture of an ISP using a NetScaler appliance consists of subscribers in private address spaces accessing the Internet through a NetScaler appliance deployed in ISP's core network. IPv4 subscribers are connected to a CPE device that includes the DS-Lite B4 functionality. The CPE device is connected to the ISP core network through ISP's IPv6-only access network. The NetScaler appliance contains the DS-Lite AFTR and LSN functionality.

IPv4 subscribers connected to the CPE device are assigned private IPv4 addresses either manually or through DHCP server running on the CPE device. On the CPE device, the AFTR tunnel endpoint address is specified manually or through DHCPv6. Configuration of CPE devices is vendor specific and therefore outside the scope of this documentation.

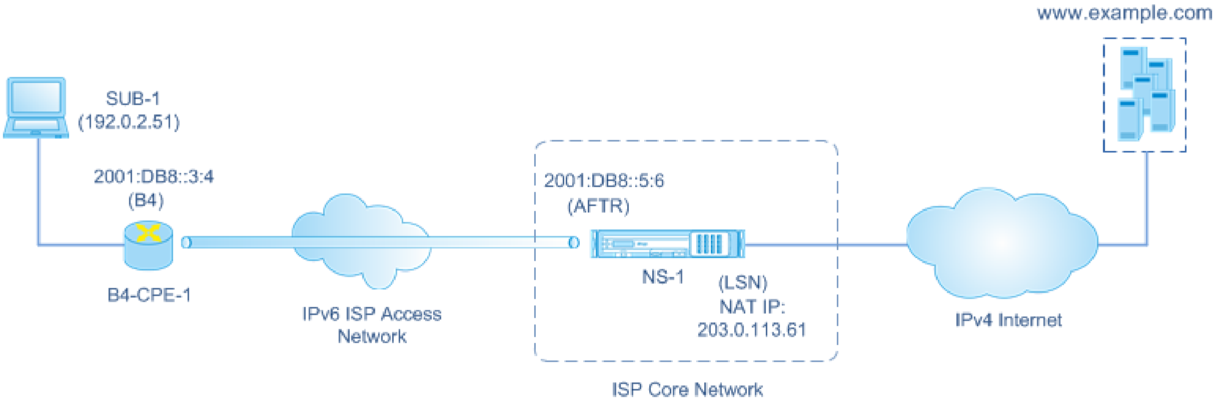
Upon receiving a request packet that is from an IPv4 subscriber and destined to a location on the Internet, the B4 component of the CPE device encapsulates the IPv4 packet in an IPv6 packet and sends it to the NetScaler appliance in the ISP core network. The NetScaler appliance's AFTR functionality decapsulates the IPv6 packet to recover the subscriber's original IPv4 packet. The LSN functionality of the NetScaler appliance translates the source IP address and port of the IPv4 packet to an NAT IP address and NAT port selected from the configured NAT pool, and then sends the packet to its destination on the Internet.

The appliance maintains a record of all active sessions that use the AFTR and LSN functionalities. These sessions are called DS-Lite sessions. The NetScaler appliance also maintains the mappings between B4 IPv6 address, subscriber IPv4 address and port, and NAT IPv4 address and port, for each DS-Lite session. These mappings are called DS-Lite LSN mappings. From DS-Lite session entries and DS-Lite LSN mapping entries, the NetScaler appliance recognizes a response packet (received from the Internet) as belonging to a particular DS-Lite session.

When the NetScaler appliance receives a response packet belonging to a particular DS-Lite session, the appliance's LSN functionality translates the destination IP address and port of the response packet from NAT IP address and port to the subscriber IP address and port, the AFTR functionality encapsulates the resulting packet in an IPv6 packet and sends it to the CPE device. The B4 functionality of the CPE device decapsulates the IPv6 packet to recover the IPv4 response packet, and then sends the IPv4 packet to the subscriber.

Example

Consider an example of a DS-Lite deployment consisting of NetScaler NS-1 in an ISP's core network, CPE device B4-CPE-1 in a subscriber premise, and a single IPv4 subscriber SUB-1. B4-CPE-1 supports the B4 functionality of DS-Lite feature.



The following table lists the settings used in this example.

Entity	Name	Details
IPv4 address of subscriber SUB-1		192.0.2.51
IPv6 address of softwire endpoint on the B4 device (B4-CPE-1)		2001:DB8::3:4
IPv6 address of the softwire endpoint on the AFTR device (NS-1)		2001:DB8::5:6

Settings on NetScaler appliance NS-1:

Entity	Name	Details
LSN client	LSN-DSLITE-CLIENT-1	Network6 (Identifying traffic from B4 devices) = 2001:DB8::3:0/100
LSN pool	LSN-DSLITE-POOL-1	LSN IPs (NAT IP) = 203.0.113.61 - 203.0.113.70
IPv6 Profile	LSN-DSLITE-PROFILE-1	Type = DS-LITE; IPv6 address (AFTR IPv6 address) = One of the NetScaler owned IPv6 address of type SNIP6 = 2001:DB8::5:6
LSN group	LSN-DSLITE-GROUP-1	LSN client = LSN-DSLITE-CLIENT-1; LSN pool = LSN-DSLITE-POOL-1;IPv6 profile = LSN-DSLITE-PROFILE-1

Following is the traffic flow in this example:

1. IPv4 subscriber SUB-1 sends a request to (`http://www.example.com/`). The IPv4 packet has:

   - Source IP address = 192.0.2.51
   - Source port = 2552
   - Destination IP address = 198.51.100.250
   - Destination port = 80

2. Upon receiving the IPv4 request packet, B4-CPE-1 encapsulates it in the payload of an IPv6 packet and then sends the IPv6 packet to NS-1. The IPv6 packet has:

   - Source IP address = 2001:DB8::3:4
   - Destination IP address = 2001:DB8::5:6

3. When NS-1 receives the IPv6 packet, the AFTR module decapsulates the packet by removing the IPv6 headers. The resulting packet is SUB-1's original IPv4 request packet.

4. The LSN module of NS-1 translates the source IP address and port of the packet to an NAT IP address and NAT port selected from the configured NAT pool. The translated IPv4 packet has:

   - Source IP address = 203.0.113.61

- Source port = 3002
- Destination IP address = 198.51.100.250
- Destination port = 80

5. The LSN module also creates an LSN mapping and session entry for this DS Lite session. The mapping includes the following information:

- Source IP address of the IPv6 packet (B4-CPE-1's IPv6 address) = 2001:DB8::3:4
- Source IP address of the IPv4 packet (SUB-1's IPv4 address) = 192.0.2.51
- Source port of the IPv4 packet = 2552
- NAT IP address = 203.0.113.61
- NAT port = 3002

6. NS-1 sends the resulting IPv4 packet to its destination on the Internet.

7. The server for www.example.com processes the request packet and sends a response packet. The IPv4 response packet has:

- Source IP address = 198.51.100.250
- Source port = 80
- Destination IP address = 203.0.113.61
- Destination port = 3002

8. Upon receiving the IPv4 packet, NS-1 examines the LSN mapping and session entries and finds that the IPv4 response packet belongs to a DS Lite session. The LSN module of NS-1 translates the destination IP address and port. The IPv4 packet now has:

- Source IP address = 198.51.100.250
- Source port = 80
- Destination IP address = 192.0.2.51
- Destination port = 2552

9. The AFTR module of NS-1 encapsulates the IPv4 packet in an IPv6 packet and then sends the IPv6 packet to B4-CPE-1. The IPv6 packet has:

- Source IP address = 2001:DB8::5:6
- Destination IP address = 2001:DB8::3:4

10. Upon receiving the packet, B4-CPE-1 decapsualtes the IPv6 packet by removing the IPv6 headers, and then sends the resulting IPv4 packet to CL-1.

Points to Consider before Configuring DS-Lite

September 21, 2020

Consider the following points before configuring DS-Lite on a NetScaler appliance:

1. You must understand the different components of DS-Lite, described in RFC 6333.

2. A DS-lite configuration on a NetScaler appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address AFTR component on a NetScaler appliance. For more information on NetScaler LSN feature, see Large Scale NAT.

3. For a DS-Lite configuration, the NetScaler appliance supports LSN for IPv4 packets that belong to one of the following protocols only. The NetScaler appliance drops IPv4 packets belonging to other protocols:

- TCP
- UDP
- ICMP

4. The NetScaler appliance supports the following ALGs DS-Lite:

- ICMP
- FTP
- TFTP
- Session Initiation Protocol (SIP)
- Real Time Streaming Protocol (RTSP)

Configuring DS-Lite

September 21, 2020

A DS-lite configuration on a NetScaler appliance uses the LSN commands sets. In a DS-Lite configuration, the LSN client entity specifies the IPv6 address or IPv6 network address or ACL6 rules for identifying the traffic from the B4 device. For more information on the NetScaler LSN feature, see Large Scale NAT. A DS-Lite configuration also includes an IPv6 profile, which specifies the IPv6 address (of type SNIP6) of the DS-Lite AFTR component on a NetScaler appliance.

Configuring DS-Lite on a NetScaler appliance consists of the following tasks:

Set the global LSN parameters. Global parameters include the amount of NetScaler memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.

- **Create an LSN client entity for identifying traffic from B4 CPE devices**. The LSN client entity refers to a set of DS-Lite B4 devices. The client entity includes IPv6 addresses or IPv6 network address or ACL6 rules for identifying the traffic from these B4 devices. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an LSN client entity and binding a subscriber to the LSN client entity. The configuration utility combines these two operations on a single screen.

- **Create an LSN pool and bind NAT IP addresses to it**. An LSN pool defines a pool of NAT IP addresses to be used by the NetScaler appliance to perform LSN. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The configuration utility combines these two operations on a single screen.
- **Create an LSN IP6 profile.** An LSN IP6 profile defines the IPv6 address of the DS-Lite AFTR component on the NetScaler appliance. The IPv6 address must be one of the NetScaler owned IPv6 address of type SNIP6.
- **(Optional) Create an LSN Transport Profile for a specified protocol**. An LSN transport profile defines various timeouts and limits, such as maximum LSN sessions and maximum ports usage that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called the default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
- **(Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it**. An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The configuration utility combines these two operations on a single screen.
- **Create an LSN Group and bind LSN pools, LSN IPv6 profile, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group**. An LSN group is an entity consisting of an LSN client, an LSN IPv6 profile, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only one LSN IPv6 profile can be bound to an LSN group, and an LSN IPv6 profile bound to an LSN group cannot be bound to other LSN groups. Only LSN Pools and LSN groups

with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, and LSN application profiles to the LSN group. The configuration utility combines these two operations in a single screen.

Configuration by Using the Command Line

To create an LSN client by using the command line interface:

At the command prompt, type:

```
add lsn client <clientname>

show lsn client
<!--NeedCopy-->
```

To bind an IPv6 network or an ACL6 rule to an LSN client by using the command line interface:

At the command prompt, type:

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

To create an LSN pool by using the command line interface:

At the command prompt, type:

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

show lsn pool
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn pool <poolname> <lsnip>

show lsn pool
```

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

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

To configure an LSN IPv6 profile by using the command line interface:

At the command prompt, type:

```
add lsn ip6profile <name> - type DS-Lite - network6 < ipv6_addr|*s >

show lsn ip6profile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn transportprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn appsprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To create an LSN group by using the command line interface:

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC )]
    [-portblocksize <positive_integer>] [-logging (ENABLED | DISABLED )]
     [-sessionLogging ( ENABLED | DISABLED )][-sessionSync ( ENABLED |
    DISABLED )] [-snmptraplimit<positive_integer>] [-ftp ( ENABLED |
    DISABLED )] [-pptp ( ENABLED |DISABLED )] [-sipalg ( ENABLED |
    DISABLED )] [-rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename
    <string> | -httphdrlogprofilename <string> | -appsprofilename <
    string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

Configuration by Using the Configuration Utility

To configure an LSN client and bind an IPv6 network address or an ACL6 rule by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Clients**, and add a client and then bind an IPv6 network address or an ACL6 rule to the client.

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

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

To configure an LSN IPv6 profile by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Profiles**, click the **IPv6** tab, and assign an IPv6 address for DS-Lite AFTR.

To configure an LSN transport profile by using the configuration utility:

1. Navigate to **System** > **Large Scale NAT** > **Profiles**.

2. On the details pane, click **Transport**, and then add a transport profile.

To configure an LSN application profile by using the configuration utility:

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

To configure an LSN group and bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles by using the configuration utility:

Navigate to **System** > **Large Scale NAT** > **Groups**, and add a group and then bind an LSN client, an LSN IPv6 profile, pools, transport profiles, and application profiles to the group.

```
> add lsn client LSN-DSLITE-CLIENT-1
Done
> bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
> add lsn pool LSN-DSLITE-POOL-1
Done
> bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
> add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
    DB8::5:6
Done
> add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1
Done
> add lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
<!--NeedCopy-->
```

Logging and Monitoring DS-Lite

You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The NetScaler appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted

- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
    - Destination IP address and port are not logged for Endpoint-Independent mapping.
    - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
    - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The NetScaler appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the follow-

ing conditions:

  - Destination IP address and port are not logged for Endpoint-Independent mapping.
  - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
DS-Lite session creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite session deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP

DS-Lite LSN mapping creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP
DS-Lite LSN mapping deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

Displaying Current DS-Lite Sessions

You can display the current DS-Lite sessions for detecting any unwanted or inefficient sessions on the NetScaler appliance. You can display all or some DS-Lite sessions, on the basis of selection parameters.

Configuration by Using the Command Line Interface

To display all DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
show lsn session -nattype DS-Lite
<!--NeedCopy-->
```

To display selected DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
show lsn session -nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

Example:

The following sample ouput displays all DS-Lite sessions existing on a NetScaler appliance:

```
show lsn session -nattype DS-Lite
```

```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP NatPort Proto Dir

1.  2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61 3002 TCP OUT

2.  2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61 52862 TCP OUT

3.  2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61 48116 ICMP OUT

4.  2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69 48305 TCP OUT

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

Configuration Using the Configuration Utility

To display all or selected DS-Lite sessions by using the configuration utility

1. **Navigate to System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. For displaying DS-Lite sessions on the basis of selection parameters, click **Search**.

Clearing DS-Lite Sessions

You can remove any unwanted or inefficient DS-Lite sessions from the NetScaler appliance. The appliance immediately releases the resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected DS-Lite sessions from the NetScaler appliance.

To clear all DS-Lite sessions by using the command line interface:

At the command prompt, type:

flush lsn session –nattype DS-Lite
show lsn session –nattype DS-Lite

To clear selected DS-Lite sessions by using the command line interface:

At the command prompt, type:

```
flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  - nattype DS-Lite
<!--NeedCopy-->
```

To clear all or selected DS-Lite sessions by using the configuration utility:

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. Click **Flush Sessions**.

Configuring DS-Lite Static Maps

September 21, 2020

The NetScaler appliance supports manual creation of DS-Lite LSN mappings, which contain the mapping between the following information:

- Subscriber's IP address and port, and IPv6 address of B4 device or component
- NAT IP address and port

Static DS-Lite LSN mappings are useful in cases where you want to ensure that the connections initiated to a NAT IP address and port map to the subscriber IP address and port through the specified B4 device (for example, web servers located in the internal network).

Note: This feature is supported in release 11.0 build 64.x and later.

To create a DS-Lite static LSN mapping by using the command line

At the command prompt, type:

```
add lsn static <name> <transportprotocol> <subscrIP> <subscrPort> [-td
    <positive_integer>] [-network6 <B4_ADDR>] [<natIP> [<natPort>]] [-
    destIP<ip_addr> [-dsttd <positive_integer>]]

show lsn static
<!--NeedCopy-->
```

Parameter Descriptions

add lsn static

- name

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

- transportprotocol

Protocol for the DS-Lite LSN mapping entry.

- subscrIP

IPv4 address of a subscriber for the DS-Lite LSN mapping entry.

- subscrPort

Port of the subscriber for the DS-Lite LSN mapping entry.

- Network6

IPv6 address of the B4 device or component.

- td

ID of the traffic domain to which the B4 device belongs. The IPv6 address of the B4 device is specified in the network6 paramter. If you do not specify an ID, the B4 device is assumed to be a part of the default traffic domain.

- natIP

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

- natPort

NAT port for this DS-Lite LSN mapping entry.

- destIP

Destination IP address for the DS-Lite LSN mapping entry.

- dsttd

ID of the traffic domain through which the destination IP address for this DS-Lite LSN mapping entry is reachable from the NetScaler appliance. If you do not specify an ID, the destination IP address is assumed to be reachable through the default traffic domain, which has an ID of 0.

To create a DS-Lite static LSN mapping by using the configuration utility

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

Configuring Deterministic NAT Allocation for DS-Lite

September 21, 2020

Deterministic NAT allocation for DS-Lite LSN deployments is a type of NAT resource allocation in which the NetScaler appliance pre-allocates, from the LSN NAT IP pool and on the basis of the specified port block size, an LSN NAT IP address and a block of ports to each subscriber (subscriber behind B4 device).

Note: This feature is supported in release 11.0 build 64.x and later.

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

The NetScaler appliance logs the allocated NAT IP address and the port block for a subscriber. For a connection, a subscriber can be identified by just its mapped NAT IP address and port block. For this reason, the NetScaler appliance does not log the creation or deletion of an LSN session.

A DS-Lite subscriber can have only one deterministic port block. If the entire block of ports is being used, the NetScaler appliance drops any new connection from the subscriber.

Example: Deterministic DS-Lite

In this example, a deterministic DS-Lite configuration includes four subscribers with IP addresses 192.0.17.5, 192.0.17.6, 192.0.17.7, and 192.0.17.8. These ipv4 subscribers are behind a B4 device having the IPv6 address 2001:DB8::3:4. In this configuration, the port block size is set to 20480 and LSN NAT IP address pool has IP addresses in the range 203.0.113.41-203.0.113.42.

The NetScaler appliance sequentially pre-allocates, from the LSN NAT IP pool and on the basis of the set port block size, an LSN NAT IP address and a block of ports to each subscriber. It assigns the first block of ports (1024-21503) on the beginning NAT IP address (203.0.113.41) to the beginning subscriber IP address (192.0.17.5). The next range of ports is assigned to the next subscriber, and so on, until the NAT address does not have enough ports for the next subscriber. At that point, the first port block on the next NAT IP address is assigned to the subscriber, and so on. The NetScaler logs the NAT IP address and the block of ports allocated for each subscriber.

The NetScaler appliance does not log any LSN session created or deleted for these subscribers.

The following table lists the NAT IP address and blocks of ports allocated to each subscriber in this example:

Subscriber IP address	Allocated NAT IP address	Allocated Block of Ports	IPv6 address of B4
192.0.17.5	203.0.113.41	1024 - 21503	2001:DB8::3:4
192.0.17.6	203.0.113.41	21504 - 41983	2001:DB8::3:4
192.0.17.7	203.0.113.41	41984 - 62463	2001:DB8::3:4
192.0.17.8	203.0.113.42	1024 - 21503	2001:DB8::3:4

Configuration Steps

You need to configure deterministic NAT as part of the DS-Lite configuration. For instructions on configuring DS-Lite, see Configuring DS-Lite.

While configuring DS-Lite, make sure that you:

- Set the NAT Type parameter to Deterministic when adding the LSN pool and the LSN group.
- Set the desired port block size parameter when adding the LSN group, unless you can accept the default value.

Points to Consider before Configuring Deterministic DS-Lite

Consider the following points before configuring deterministic DS-Lite:

- The complete IP address of each subscriber must be specified in a separate add lsn client command, by setting the Network and Netmask parameters. (Set Netmask to 255.255.255.255.) Also the IPv4 address of the B4 device specified in Network6 parameter must be complete (/128 prefix). In other words, Network and Network6 parameter do not accept addresses other than /32 bit mask and /128 prefix, respectively.
- The NetScaler appliance drops connections from subscribers that are not specified in any deterministic DS-Lite configuration but are behind B4 devices specified in a deterministic DS-lite configuration.
- The NetScaler appliance recognizes subscribers having the same IPv4 address as different subscribers if they are behind different B4 devices. A combination of subscriber IPv4 address and B4 device defines a unique subscriber in the LSN client entity of a DS-Lite configuration.

Sample Deterministic DS-Lite Configuration:

The following configuration uses the settings listed in section Example: Deterministic DS-Lite.

```
add lsn client LSN-DSLITE-CLIENT-10

```

```
Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.5 -netmask
    255.255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.6 -netmask 255
    .255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.7 -netmask 255
    .255.255.255 -network6 2001:DB8::3:4/128

Done
bind lsn client LSN-DSLITE-CLIENT-10 -network 192.0.17.8 -netmask 255
    .255.255.255 -network6 2001:DB8::3:4/128

Done
add lsn pool LSN-DSLITE-POOL-10 -nattype DETERMINISTIC

Done
bind lsn pool LSN-DSLITE-POOL-10  203.0.113.41-203.0.113.42

Done
add lsn ip6profile LSN-DSLITE-PROFILE-10 -type DS-Lite -network6 2001:
    DB8::5:6

Done
add lsn group LSN-DSLITE-GROUP-10 -clientname LSN-DSLITE-CLIENT-10 -
    nattype DETERMINISTIC -portblocksize 20480 -ip6profile LSN-DSLITE-
    PROFILE-10

Done
bind lsn group LSN-DSLITE-GROUP-10 -poolname LSN-DSLITE-POOL-10

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

Configuring Application Layer Gateways for DS-Lite

September 21, 2020

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

The NetScaler appliance supports ALG for the following protocols for DS-Lite:

- FTP
- ICMP
- TFTP
- SIP
- RTSP

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 21, 2020

You can enable or disable ALG for the FTP protocol for a DS-Lite configuration by enabling or disabling the FTP ALG option of the LSN group of the configuration.

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

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

Application Layer Gateway for SIP Protocol

September 21, 2020

Using DS-Lite with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. SIP ALG for DS-Lite is compliant with RFC 3261, RFC 3581, RFC 4566, and RFC 4475.

Limitations of SIP ALG

SIP ALG for DS-Lite has the following limitations:

- Only SDP payload is supported.

- The following are not supported:
    - Multicast IP addresses
    - Encrypted SDP
    - SIP TLS
    - FQDN translation
    - SIP layer authentication
    - Admin partitions
    - NetScaler clusters
    - Multipart body
    - Line folding

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring DS-Lite. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:

o IP Pooling = PAIRED

o Address and Port Mapping = ENDPOINT-INDEPENDENT

o Filtering = ENDPOINT-INDEPENDENT

- Create a SIP ALG profile and make sure that you define either the source port range or destination port range. Bind the SIP ALG profile to the LSN group
- Enable SIP ALG in the LSN group

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string>[-sipalg ( ENABLED | DISABLED )]

show lsn group<groupname>
<!--NeedCopy-->
```

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn sipalgprofile<sipalgprofilename>[-dataSessionIdleTimeout<
    positive_integer>][-sipSessionTimeout<positive_integer>][-
    registrationTimeout<positive_integer>][-sipsrcportrange<port[-port
    ]>][-sipdstportrange<port[-port]>][-openRegisterPinhole ( ENABLED |
    DISABLED )][-openContactPinhole ( ENABLED | DISABLED )][-
    openViaPinhole ( ENABLED | DISABLED )][-openRecordRoutePinhole (
    ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP )[-
    openRoutePinhole ( ENABLED | DISABLED )][-rport ( ENABLED | DISABLED
     )]

show lsn sipalgprofile<sipalgprofilename>
<!--NeedCopy-->
```

Sample Configuration

The following sample DS-Lite configuration, SIP ALG is enabled for TCP traffic from B4 devices in the network 2001:DB8::3:0/96.

```
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/96
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
    DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-1 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn sipalgprofile SIPALGPROFILE-1  -sipdstportrange 5060 -
    sipTransportProtocol TCP
Done
add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -
    portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1 -sipalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -appsprofilename LSN-DSLITE-APPS-
    PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -sipalgprofilename SIPALGPROFILE-1
```

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

Application Layer Gateway for RTSP Protocol

September 21, 2020

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. NetScaler functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to NetScaler owned public IP address called LSN IP address.
- Inbound response—LSN IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

Limitations of RTSP ALG

The RTSP ALG does not support the following:

- Multicast RTSP sessions
- RTSP session over UDP
- Admin partitions
- NetScaler clusters
- RTSP Authentication
- HTTP tunneling

Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring DS-Lite. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:
    - IP Pooling = PAIRED
    - Address and Port Mapping = ENDPOINT-INDEPENDENT
    - Filtering = ENDPOINT-INDEPENDENT
- Enable RTSP ALG in the LSN group
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

Sample RTSP ALG Configuration

The following sample DS-Lite configuration, RTSP ALG is enabled for TCP traffic from B4 devices in the network 2001:DB8::4:0/96.

Sample RTSP ALG Configuration:

```
add lsn client LSN-DSLITE-CLIENT-5
Done
bind lsn client LSN-DSLITE-CLIENT-5 -network6 2001:DB8::4:0/96
Done
add lsn pool LSN-DSLITE-POOL-5
Done
bind lsn pool LSN-DSLITE-POOL-5 203.0.113.61 - 203.0.113.70
Done
```

```
add lsn ip6profile LSN-DSLITE-PROFILE-5 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn appsprofile LSN-DSLITE-APPS-PROFILE-5 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-5 -rtspIdleTimeout 1000 -rtspportrange 554
Done
add lsn group LSN-DSLITE-GROUP-5 -clientname LSN-DSLITE-CLIENT-5 -portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-5 -rtspalg ENABLED
Done
bind lsn group LSN-DSLITE-GROUP-5 -poolname LSN-DSLITE-POOL-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -appsprofilename LSN-DSLITE-APPS-PROFILE-5
Done
bind lsn group LSN-DSLITE-GROUP-5 -rtspalgprofilename RTSPALGPROFILE-5
Done
<!--NeedCopy-->
```

Logging and Monitoring DS-Lite

September 21, 2020

You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The NetScaler appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber’s IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name

- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
    - Destination IP address and port are not logged for Endpoint-Independent mapping.
    - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.
    - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.You can log DS-Lite information to diagnose or troubleshoot problems, and to meet legal requirements. The NetScaler appliance supports all LSN logging features for logging DS-Lite information. For configuring DS-Lite logging, use the procedures for configuring LSN logging, described at Logging and Monitoring LSN.

A log message for a DS-Lite LSN mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (MAPPING)
- Whether the DS-Lite LSN mapping entry was created or deleted
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
    - Destination IP address and port are not logged for Endpoint-Independent mapping.
    - Only the destination IP address is logged for Address-Dependent mapping. The port is not logged.

  - Destination IP address and port are logged for Address-Port-Dependent mapping.

A log message for a DS-Lite session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the DS-Lite session is created or removed
- IPv6 address of B4
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table shows sample DS-Lite log entries of each type stored on the configured log servers. These log entries are generated by a NetScaler appliance whose NSIP address is 10.102.37.115.

LSN Log Entry Type	Sample Log Entry
DS-Lite session creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:38 GMT 0-PPE-1 : default LSN LSN_SESSION 37647607 0 : SESSION CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol:TCP
DS-Lite session deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:22 GMT 0-PPE-1 : default LSN LSN_SESSION 37647617 0 : SESSION DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 203.0.113.61:3002, Destination IP:Port:TD 198.51.100.250:80:0, Protocol: TCP
DS-Lite LSN mapping creation	Local4.Informational 10.102.37.115 08/14/2015:13:35:39 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647610 0 : EIM CREATED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

DS-Lite LSN mapping deletion	Local4.Informational 10.102.37.115 08/14/2015:13:38:25 GMT 0-PPE-1 : default LSN LSN_EIM_MAPPING 37647618 0 : EIM DELETED 2001:DB8::3:4 Client IP:Port:TD 192.0.2.51:2552:0, NatIP:NatPort 198.51.100.250:80, Protocol: TCP

Displaying Current DS-Lite Sessions

You can display the current DS-Lite sessions for detecting any unwanted or inefficient sessions on the NetScaler appliance. You can display all or some DS-Lite sessions, on the basis of selection parameters.

To display all DS-Lite sessions by using the command line interface

At the command prompt, type:

```
show lsn session -nattype DS-Lite
<!--NeedCopy-->
```

To display selected DS-Lite sessions by using the command line interface

At the command prompt, type:

```
show lsn session -nattype DS-Lite [-clientname <string>] [-network <ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

The following sample ouput displays all DS-Lite sessions existing on a NetScaler appliance:

> show lsn session –nattype DS-Lite

```
  B4-Address SubscrIP SubscrPort SubscrTD DstIP DstPort DstTD NatIP NatPort Proto Dir

1. 2001:DB8::3:4 192.0.2.51 2552 0 198.51.100.250 80 0 203.0.113.61 3002 TCP OUT

2. 2001:DB8::3:4 192.0.2.51 3551 0 198.51.100.300 80 0 203.0.113.61 52862 TCP OUT
```

```

3. 2001:DB8::3:4 192.0.2.100 4556 0 198.51.100.250 0 0 203.0.113.61
    48116 ICMP OUT

4. 2001: DB8::190 192.0.2.150 3881 0 198.51.100.199 80 0 203.0.113.69
    48305 TCP OUT
Done
<!--NeedCopy-->
```

Configuration Using the Configuration Utility

To display all or selected DS-Lite sessions by using the configuration utility

1. **Navigate to System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. For displaying DS-Lite sessions on the basis of selection parameters, click **Search**.

Clearing DS-Lite Sessions

You can remove any unwanted or inefficient DS-Lite sessions from the NetScaler appliance. The appliance immediately releases the resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected DS-Lite sessions from the NetScaler appliance.

To clear all DS-Lite sessions by using the command line interface

At the command prompt, type:

```
flush lsn session  - nattype DS-Lite

show lsn session  - nattype DS-Lite
<!--NeedCopy-->
```

To clear selected DS-Lite sessions by using the command line interface

At the command prompt, type:

```
flush lsn session  - nattype DS-Lite [-clientname <string>] [-network <
    ip_addr> [-netmask <netmask>] [-td <positive_integer>]] [-natIP <
    ip_addr> [-natPort <port>]]

show lsn session  - nattype DS-Lite
```

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

To clear all or selected DS-Lite sessions by using the configuration utility

1. Navigate to **System** > **Large Scale NAT** > **Sessions**, and click the **DS-Lite** tab.
2. Click **Flush Sessions**.

Logging HTTP Header Information

The NetScaler appliance can log request header information of an HTTP connection that is using the DS-Lite functionality. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to
- HTTP Method specified in the HTTP request
- HTTP version used in the HTTP request
- IPv4 address of the subscriber that sent the HTTP request

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set of subscribers. For example, an ISP can use this feature to find out the most popular website among a set of subscribers.

Configuration Steps

Perform the following tasks for configuring the NetScaler appliance to log HTTP header information:

- **Create an HTTP header log profile**. An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging.
- **Bind the HTTP header to an LSN group of a DS-Lite LSN configuration**. Bind the HTTP header log profile to an LSN group of an LSN configuration by setting the HTTP header log profile name parameter to the name of the created HTTP header log profile. The NetScaler appliance then logs HTTP header information of any HTTP requests related to the LSN group. An HTTP header log profile can be bound to multiple LSN groups, but an LSN group can have only one HTTP header log profile.

To create an HTTP header log profile by using the command line interface

At the command prompt, type:

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

```

```
show lsn httphdrlogprofile
<!--NeedCopy-->
```

To bind an HTTP header log profile to an LSN group by using the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

Sample Configuration

In the following DS-Lite LSN configuration, HTTP header log profile HTTP-Header-LOG-1 is bound to LSN group LSN-DSLITE-GROUP-1. The log profile has all the HTTP attributes (URL, HTTP method, HTTP version, and HOST IP address) enabled for logging, so that all these attributes are logged for any HTTP requests from B4 devices (in the network 2001:DB8:5001::/96).

Sample Configuration:

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1

Done

add lsn client LSN-DSLITE-CLIENT-1

Done

bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100

Done

add lsn pool LSN-DSLITE-POOL-1

Done

bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70

Done

add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:
    DB8::5:6
```

```

Done

add lsn group LSN-DSLITE-GROUP-1 -clientname LSN-DSLITE-CLIENT-1 -portblocksize 1024 -ip6profile LSN-DSLITE-PROFILE-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-DSLITE-POOL-1

Done

bind lsn group LSN-DSLITE-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG-1

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

IPFIX Logging

The NetScaler appliance supports sending information about LSN events in Internet Protocol Flow Information Export (IPFIX) format to the configured set of IPFIX collector(s). The appliance uses the existing AppFlow feature to send LSN events in IPFIX format to the IPFIX collectors.

IPFIX based logging is available for the following DS_Lite related events:

- Creation or deletion of an LSN session.
- Creation or deletion of an LSN mapping entry.
- Allocation or de-allocation of port blocks in the context of deterministic NAT.
- Allocation or de-allocation of port blocks in the context of dynamic NAT.
- Whenever subscriber session quota is exceeded.

Points to Consider before you Configure IPFIX logging

Before you start configuring IPSec ALG, consider the following points:

- You must configure the AppFlow feature and IPFIX collector(s) on the NetScaler appliance. For instructions, see Configuring the AppFlow feature.

Configuration Steps

Perform the following tasks for logging LSN information in IPFIX format:

- **Enable LSN logging in the AppFlow configuration**. Enable the LSN logging parameter as part of AppFlow configuration.
- **Create an LSN log profile**. An LSN log profile includes the IPFIX parameter that enables or disables the log information in IPFIX format.
- **Bind the LSN log profile to an LSN group of an LSN configuration**. Bind the LSN log profile to one or multiple LSN group(s). Events related to the bound LSN group will be logged in IPFIX format.

To enable LSN logging in the AppFlow configuration by using the CLI

At the command prompt, type:

```
set appflow param -lsnLogging (ENABLED |DISABLED )

show appflow param
<!--NeedCopy-->
```

To create an LSN log profile by using the CLIAt the command prompt, type

At the command prompt, type:

```
set lsn logprofile <logProfileName>  -logipfix ( ENABLED | DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind the LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

To create an LSN log profile by using the GUI

Navigate to **System** > **Large Scale NAT** > **Profiles**, click **Log** tab, and then add a log profile.

To bind the LSN log profile to an LSN group of an LSN configuration by using the GUI

1. Navigate to **System** > **Large Scale NAT** > **LSN Group,** open the **LSN** group.
2. In **Advanced Settings,** click **+ Log Profile** to bind the created Log profile to the LSN group.

Port Control Protocol for DS-Lite

September 21, 2020

NetScaler appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A NetScaler appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- (Optional) Create a PCP profile. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:
  - Mapping: Enabled
  - Peer: Enabled
  - Minimum map life: 120 seconds

  - Maximum max life: 86400 seconds
  - Announce count: 10
  - Third Party: Disabled
- Create a PCP server and bind a PCP profile to it. Create a PCP server on the NetScaler appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- Bind the PCP server to an LSN group of an LSN configuration. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group. Note: A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

Sample Configuration for DS-LITE

In the following sample configuration, PCP server PCP-SERVER-1, with PCP settings from PCP-DSLITE-PROFILE-1, is bound to LSN group LSN-DSLITE-GROUP-1. PCP-SERVER-9 serves PCP requests from IPv4 subscribers behind B4 devices from network 2001:DB8::3:0/100.

Sample configuration:

```
add pcp profile PCP-DSLITE-PROFILE-1 -minMapLife 300
Done
add pcp server PCP-DSLITE-SERVER-1 192.0.3.10  -pcpProfile PCP-DSLITE-PROFILE-1
Done
add lsn client LSN-DSLITE-CLIENT-1
Done
bind lsn client LSN-DSLITE-CLIENT-1 -network6 2001:DB8::3:0/100
Done
add lsn pool LSN-DSLITE-POOL-1
Done
bind lsn pool LSN-DSLITE-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-DSLITE-PROFILE-1 -type DS-Lite -network6 2001:DB8::5:6
Done
add lsn group LSN-DSLITE-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-DSLITE-GROUP-1 -poolname  PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

Large Scale NAT64

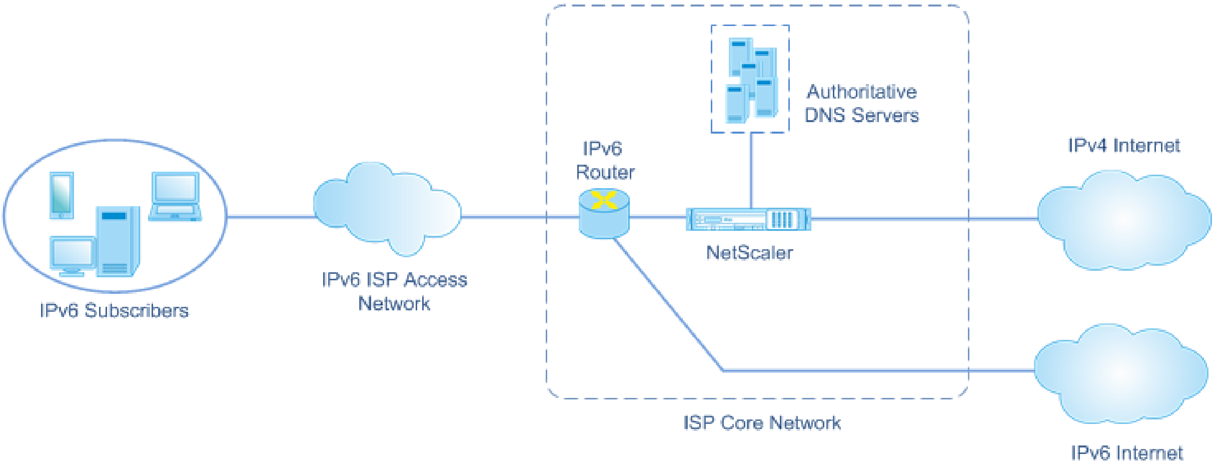
September 21, 2020

Because of the imminent exhaustion of IPv4 addresses, ISPs have started transitioning to IPv6 infrastructure. But during the transition, ISPs must continue to support IPv4 along with IPv6, because most of the public Internet still uses IPv4. Large scale NAT64 is an IPv6 transition solution for ISPs with IPv6 infrastructure to connect their IPv6-only subscribers to the IPv4 Internet. DNS64 is a solution for enabling discovery of IPv4-only domains by IPv6-only clients. DNS64 is used with large scale NAT64 to enable seamless communication between IPv6-only clients and IPv4-only servers.

A NetScaler appliance implements large scale NAT64 and DNS64 and is compliant with RFCs 6145, 6146, 6147, 6052, 3022, 2373, 2765, and 2464.

Architecture

The NAT64 architecture of an ISP using a NetScaler appliance consists of IPv6 subscribers accessing the IPv4 Internet through a NetScaler appliance deployed in the ISP's core network. IPv6 subscribers are connected to the ISP core network through the ISP's IPv6-only access network.



The large scale NAT64 functionality of a NetScaler appliance enables communication between IPv6 clients and IPv4 servers through IPv6-to-IPv4 packet translation, and vice versa, while maintaining session information on the NetScaler appliance.NetScaler DNS64 functionality represents IPv4-only domains to IPv6-subscribers by synthesizing DNS AAAA records for IPv4-only domains and sending them to the subscribers.

Large scale NAT64 has two main components: NAT64 prefix and NAT IPv4 pool. DNS64 has one main component, DNS64 prefix, which has the same value as NAT64 prefix.

Upon receiving an AAAA request from an IPv6-only subscriber for a domain name that is hosted on an IPv4-only web server on the Internet, the NetScaler DNS64 functionality synthesizes an AAAA record for the domain name and sends it to the subscriber. The AAAA record is synthesized by concatenating the DNS64 prefix (which is set to the NAT64 prefix) and the actual IPv4 address of the domain name.

The subscriber now has an IPv6 destination address that corresponds to the desired domain name. The subscriber sends the request to the synthesized IPv6 address. Upon receiving the IPv6 request, the large scale NetScaler NAT64 functionality translates the IPv6 request packet to an IPv4 request packet. Large scale NAT64 sets the IPv4 request's destination address to the IPv4 address, which is extracted from the IPv6 request's destination address by stripping the NAT64 prefix from the IPv6 address. The destination port is retained from the IPv6 request. Large Scale NAT64 also sets the source IP address:source port of the IPv4 packet to the NAT IP address:NAT port selected from the configured NAT pool.

The appliance maintains a record of all active sessions that use the large scale NAT64 functionality. These sessions are called large scale NAT64 sessions. The appliance also maintains the mappings between subscriber IPv6 address and port, and NAT IPv4 address and port, for each large scale NAT64 session. These mappings are called large scale NAT64 mappings. From large scale NAT64 session entries and large scale NAT64 mapping entries, the NetScaler appliance recognizes a response packet (received from the Internet) as belonging to a particular NAT64 session.

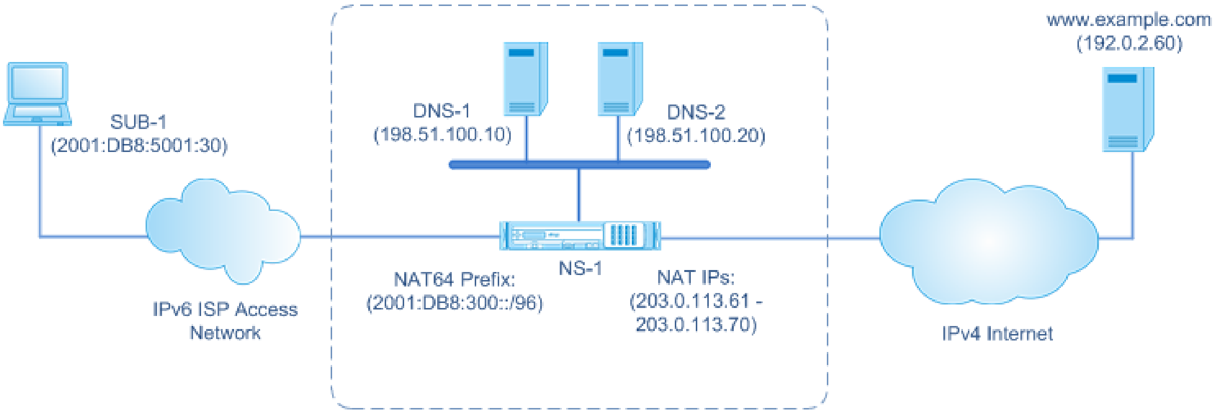
When the appliance receives an IPv4 response packet belonging to a particular NAT64 session, it uses the information stored in the NAT64 session to translate the IPv4 packet into an IPv6 packet, and then sends the IPv6 response packet to the subscriber.

Example: Traffic Flow of NAT64 and DNS64 Deployment

Consider an example of a large scale NAT64 and DNS64 deployment consisting of NetScaler appliance NS-1 and two local DNS servers, DNS-1 and DNS-2, in an ISP's core network, and IPv6 subscriber SUB-1. SUB-1 is connected to NS-1 through the ISP's IPv6 access network. NS-1 includes large scale NAT64 and DNS64 configurations for enabling the communication between IPv6 subscriber SUB-1 and IPv4 hosts (internal and external).

Large scale NAT64 configuration includes a NAT64 prefix (2001:DB8:300::/96) and NAT IPv4 pool for translation of IPv6 requests to IPv4 requests and IPv4 responses to IPv6 responses.

DNS64 configuration includes a DNS load balancing virtual server LBVS-DNS64-1 (2001:DB8:9999::99) and a DNS64 prefix (2001:DB8:300::/96). LBVS-DNS64-1 represents local DNS server DNS-1 and DNS-2 to ISP's subscribers. The DNS64 prefix, which has the same value as the NAT64 prefix, is used for synthesizing DNS AAAA records from DNS A records received from DNS servers DNS-1 and DNS-2. NS-1 responds with a synthesized AAAA record to SUB-1 for a DNS request to resolve an IPv4 host.



DNS64 Traffic Flow

Traffic flows between IPv6 subscriber SUB-1 and site `www.example.com`, which resides on an IPv4-only web server on the Internet, as follows:

1. IPv6 subscriber SUB-1 sends a DNS AAAA request for `www.example.com` to its designated DNS server (2001:DB8:9999::99).

2. DNS load balancing virtual server LBVS-DNS64-1 (2001:DB8:9999::99) on NetScaler appliance NS1 receives the AAAA request. LBVS-DNS64-1's load balancing algorithm selects DNS server DNS-1 and forwards the AAAA request to it.
3. DNS-1 returns an empty record or an error message, because there is no AAAA record available for `www.example.com`.
4. Because the DNS64 option is enabled on LBVS-DNS64-1 and the AAAA request from CL1 matches the condition specified in DNS64-Policy-1, NS1 sends a DNS A request to DNS-1 for the IPv4 address of `www.example.com`.
5. DNS-1 responds with the A record of 192.0.2.60 for `www.example.com`.
6. DNS64 module on NS1 synthesizes an AAAA record for `www.example.com` by concatenating the DNS64 Prefix (2001:DB8:300::/96) associated with LBVS-DNS64-1, and IPv4 address (192.0.2.60) for `www.example.com` = 2001:DB8:300::192.0.2.60
7. NS1 sends the synthesized AAAA record to IPv6 client CL1. NS1 also caches the A record into its memory. NS1 uses the cached A record to synthesize AAAA records for subsequent AAAA requests.

NAT64 Traffic Flow

1. IPv6 subscriber SUB-1 sends a request to 2001:DB8:5001:30 `www.example.com`. The IPv6 packet has:

- Source IP address = 2001:DB8:5001:30
- Source port = 2552
- Destination IP address = 2001:DB8:300::192.0.2.60
- Destination port = 80

2.IPv6 subscriber SUB-1 sends a request to 2001:DB8:5001:30 `www.example.com`. The IPv6 packet has:

- Source IP address = 2001:DB8:5001:30
- Source port = 2552
- Destination IP address = 2001:DB8:300::192.0.2.60
- Destination port = 80

3. When NS-1 receives the IPv6 packet, the large scale NAT64 module creates a translated IPv4 request packet with:

- Source IP address = One of the IPv4 addresses available in the configured NAT pool (203.0.113.61)
- Source port = One of ports available with the allocated NAT IPv4 address (3002)
- Destination IP address = IPv4 address extracted from the IPv6 request's destination address by stripping the NAT64 prefix (2001:DB8:300::/96) from the IPv6 address (192.0.2.60)
- Destination port = IPv6 request's destination port (80)

4. The large scale NAT64 module also creates mapping and session entries for this large scale NAT64 flow. The session and mapping entries include the following information:

- Source IP address of the IPv6 packet = 2001:DB8:5001:30
- Source port of the IPv6 packet = 2552
- NAT IP address = 203.0.113.61
- NAT port = 3002
- NS-1 sends the resulting IPv4 packet to its destination on the Internet.

5. Upon receiving the request packet, the server for `www.example.com` processes the packet and sends a response packet to NS-1. The IPv4 response packet has:

- Source IP address = 192.0.2.60
- Source port = 80
- Destination IP address = 203.0.113.61
- Destination port = 3002

6. Upon receiving the IPv4 response packet, NS-1 examines the large scale NAT64 mapping and session entries and finds that the IPv4 response packet belongs to a large scale NAT64 session. The large scale NAT64 module creates a translated IPv6 response packet:

- Source IP address = 2001:DB8:300::192.0.2.60
- Source port = 80
- Destination IP address = 2001:DB8:5001:30
- Destination port = 2552

7. NS-1 sends the translated IPv6 response to client SUB-1.

Large Scale NAT64 features Supported on NetScaler appliances

Large scale NAT64 on a NetScaler appliance supports the standard LSN feature set. For more information on these LSN features, see LSN Architecture.

Following are some of the large scale NAT64 features supported on NetScaler appliances:

- ALGs. Support of application Layer Gateway (ALG) for SIP, RTSP, FTP, ICMP, and TFTP protocols.
- Deterministic/Fixed NAT. Support for pre-allocation of blocks of ports to subscribers to minimize logging.
- Mapping. Support of Endpoint-independent mapping (EIM), Address-dependent mapping (ADM), and Address-Port dependent mapping (APDM).
- Filtering. Support of Endpoint-Independent Filtering (EIF), Address-Dependent Filtering (ADF), and Address-Port-Dependent Filtering (APDF).
- Quotas. Configurable limits on number of ports, sessions per subscriber, and sessions per LSN group.
- Static Mapping. Support for manually defining a large scale NAT64 mapping.

- Hairpin Flow. Support for communication between subscribers or internal hosts using NAT IP addresses.
- 464XLAT connections. Support for communication between IPv4-only applications on IPv6 subscriber hosts and IPv4 hosts on the Internet through IPv6 network.
- Variable length NAT64 and DNS64 prefixes. The NetScaler appliance supports defining NAT64 and DNS64 prefixes of lengths of 32, 40, 48, 56, 64, and 96.
- Multiple NAT64 and DNS64 prefix. The NetScaler appliance supports multiple NAT64 and DNS64 prefixes.
- LSN Clients. Support for specifying or identifying subscribers for large scale NAT64 by using IPv6 prefixes and extended ACL6 rules.
- Logging. Support for logging NAT64 sessions for law enforcement. In addition, the following are also supported for logging.
  - **Reliable SYSLOG**. Support for sending SYSLOG messages over TCP to external log servers for a more reliable transport mechanism.
  - **Load balancing of log servers**. Support for load balancing of external log servers for preventing storage of redundant log messages.
  - **Minimal Logging**. Deterministic LSN configurations or Dynamic LSN configurations with port block significantly reduce the large scale NAT64 log volume.
  - **Logging MSISDN information**. Support for including subscribers' MSISDN information in large scale NAT64 logs to identify and track subscriber activity over the Internet.

Points to Consider for Configuring Large Scale NAT64

September 21, 2020

Before you start configuring large scale NAT64 and DNS64, consider these points:

1. Make sure you understand the different components of large scale NAT64, described in RFCs.
2. The NetScaler appliance supports only the following ALGs for large Scale NAT64:
   - FTP
   - TFTP
   - ICMP
   - SIP
   - RTSP
3. In a high availability setup of two NetScaler appliances, large NAT64 session synchronization (connection mirroring) is not supported.

Configuring DNS64

September 21, 2020

Creating the required entities for stateful NAT64 configuration on the NetScaler appliance involves the following procedures:

- Add DNS services. DNS services are logical representations of DNS servers for which the NetScaler appliance acts as a DNS proxy server. For more information on setting optional parameters of a service, see "Load Balancing".
- Add DNS64 action and DNS64 policy and then bind the DNS64 action to the DNS64 policy. A DNS64 policy specifies conditions to be matched against traffic for DNS64 processing according to the settings in the associated DNS64 action. The DNS64 action specifies the mandatory DNS64 prefix and the optional exclude-rule and mapped-rule settings.
- Create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it. The DNS load balancing virtual server acts as a DNS proxy server for DNS servers represented by the bound DNS services. Traffic arriving at the virtual server is matched against the bound DNS64 policy for DNS64 processing. For more information on setting optional parameters of a load balancing virtual server, see "Load Balancing.

> **Note**
>
> The command line interface has separate commands for these two tasks, but the GUI combines them in a single dialog box.

- Enable caching of DNS records. Enable the global parameter for the NetScaler appliance to cache DNS records, which are obtained through DNS proxy operations. For more information on enabling caching of DNS records, see "Enabling Caching of DNS Records".

To create a service of type DNS by using the command line interface

At the command prompt, type:

```
add service <name> <IP> <serviceType> <port> …
<!--NeedCopy-->
```

To create a DNS64 action by using the command line interface

At the command prompt, type:

```
add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <expression>] [-excludeRule <expression>]
<!--NeedCopy-->
```

To create a DNS64 policy by using the command line interface

At the command prompt, type:

```
add dns policy64 <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

To create a DNS load balancing virtual server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED ) [-bypassAAAA ( YES | NO)] …
<!--NeedCopy-->
```

To bind the DNS services and the DNS64 policy to the DNS load balancing virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName> ...

bind lb vserver <name> -policyName <string> -priority <positive_integer> ...
<!--NeedCopy-->
```

Sample configuration:

```
add service SVC-DNS-1 203.0.113.50 DNS 53
Done
add service SVC-DNS-2 203.0.113.60 DNS 53
Done
add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
Done
add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:DB8:5001::/64)" -action DNS64-Action-1
Done
add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-1
Done
bind lb vserver LBVS-DNS64-1 SVC-DNS-2
Done
bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
```

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

Configuring Large Scaler NAT64

September 21, 2020

A large scale NAT64 configuration on a NetScaler appliance uses the LSN commands sets. In a large scale NAT64 configuration, the LSN client entity specifies the IPv6 address or IPv6 network address, or ACL6 rules, for identifying IPv6 subscribers. A NAT64 configuration also includes an IPv6 profile, which specifies a NAT64 prefix.

Configuring NAT64 on a NetScaler appliance consists of the following tasks:

- Set the global LSN parameters. Global parameters include the amount of NetScaler memory reserved for the LSN feature and synchronization of LSN sessions in a high availability setup.
- Create an LSN client entity for identifying traffic from IPv6 subscribers. The LSN client entity refers to a set of IPv6 subscribers. The client entity includes IPv6 addresses or IPv6 network prefixes, or ACL6 rules, for identifying the traffic from these subscribers. An LSN client can be bound to only one LSN group. The command line interface has two commands for creating an LSN client entity and binding a subscriber to the LSN client entity. The GUI combines these two operations on a single screen.
- Create an LSN pool and bind NAT IP addresses to it. An LSN pool defines a pool of NAT IP addresses to be used by the NetScaler appliance to perform large scale NAT64. The command line interface has two commands for creating an LSN pool and binding NAT IP addresses to the LSN pool. The GUI combines these two operations on a single screen.
- Create an LSN IP6 profile. An LSN IP6 profile defines the NAT64 prefix for a large scale NAT64 configuration.
- (Optional) Create an LSN Transport Profile for a specified protocol. An LSN transport profile defines various timeouts and limits, such as maximum large scale NAT64 sessions and maximum ports usage that a subscriber can have for a given protocol. You bind an LSN transport profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. A profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN transport profile with default settings for TCP, UDP, and ICMP protocols is bound to an LSN group during its creation. This profile is called the default transport profile. An LSN transport profile that you bind to an LSN group overrides the default LSN transport profile for that protocol.
- (Optional) Create an LSN Application Profile for a specified protocol and bind a set of destination ports to it. An LSN application profile defines the LSN mapping and LSN filtering controls of a group for a given protocol and for a set of destination ports. For a set of destination ports, you

bind an LSN profile for each protocol (TCP, UDP, and ICMP) to an LSN group. A profile can be bound to multiple LSN groups. An LSN application profile bound to an LSN group applies to all subscribers of an LSN client bound to the same group. By default, one LSN application profile with default settings for TCP, UDP, and ICMP protocols for all destination ports is bound to an LSN group during its creation. This profile is called a default application profile. When you bind an LSN application profile, with a specified set of destination ports, to an LSN group, the bound profile overrides the default LSN application profile for that protocol at that set of destination ports. The command line interface has two commands for creating an LSN application profile and binding a set of destination ports to the LSN application profile. The GUI combines these two operations on a single screen.

- Create an LSN Group and bind LSN pools, LSN IPv6 profile, (optional) LSN transport profiles, and (optional) LSN application profiles to the LSN group. An LSN group is an entity consisting of an LSN client, an LSN IPv6 profile, LSN pool(s), LSN transport profile(s), and LSN application profiles(s). A group is assigned parameters, such as port block size and logging of LSN sessions. The parameter settings apply to all the subscribers of an LSN client bound to the LSN group. Only one LSN IPv6 profile can be bound to an LSN group, and an LSN IPv6 profile bound to an LSN group cannot be bound to other LSN groups. Only LSN Pools and LSN groups with the same NAT type settings can be bound together. Multiples LSN pools can be bound to an LSN group. Only one LSN client entity can be bound to an LSN group, and an LSN client entity bound to an LSN group cannot be bound to other LSN groups. The command line interface has two commands for creating an LSN group and binding LSN pools, LSN transport profiles, and LSN application profiles to the LSN group. The GUI combines these two operations in a single screen.

Configuration Using the Command Line

You can create different configurations using the command line interface. Follow the steps given below.

To create an LSN client by using the command line interface

At the command prompt, type:

```
add lsn client <clientname

show lsn client
<!--NeedCopy-->
```

To bind an IPv6 network or an ACL6 rule to an LSN client by using the command line interface

At the command prompt, type:

```
bind lsn client <clientname> (-network6 <ipv6_addr|*>| -acl6name <
    string>)

show lsn client
<!--NeedCopy-->
```

To create an LAN pool by using the command line interface

At the command prompt, type:

```
add lsn pool <poolname>

show lsn pool <poolname>
<!--NeedCopy-->
```

To bind NAT IP addresses to an LSN pool by using the command line interface

At the command prompt, type:

```
bind lsn pool <poolname> <lsnip>

show lsn pool
<!--NeedCopy-->
```

> **Note**
>
> For removing NAT IP (LSN IP addresses) addresses from an LSN pool, use the unbind lsn pool command.

To configure an LSN IPv6 profile by using the command line interface

At the command prompt, type:

```
add lsn ip6profile <name>  - type NAT64 -natprefix <ipv6_addr|*>

show lsn ip6profile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn transportprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

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

show lsn appsprofile
<!--NeedCopy-->
```

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

At the command prompt, type:

```
bind lsn appsprofile <appsprofilename> <lsnport>

show lsn appsprofile
<!--NeedCopy-->
```

To create an LSN group by using the command line interface

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-nattype ( DYNAMIC |
    DETERMINISTIC )] [-portblocksize <positive_integer>] [-logging(
    ENABLED | DISABLED )] [-sessionLogging ( ENABLED | DISABLED )][-
    sessionSync ( ENABLED | DISABLED )] [-snmptraplimit<positive_integer
    >] [-ftp ( ENABLED | DISABLED )] [-sipalg ( ENABLED | DISABLED )] [-
    rtspalg ( ENABLED |DISABLED )] [-ip6profile <string>]

show lsn group
```

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

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

At the command prompt, type:

```
bind lsn group <groupname> (-poolname <string> | -transportprofilename <string> | -httphdrlogprofilename <string> | -appsprofilename <string> | -sipalgprofilename <string> | rtspalgprofilename <string>)

show lsn group
<!--NeedCopy-->
```

Sample Large Scale NAT64 Configurations

Here are some sample configurations of large scale NAT64:

Simple large scale NAT64 configuration with default settings:

```
add lsn client LSN-NAT64-CLIENT-1

bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96

add lsn pool LSN-NAT64-POOL-1

bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70

add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8:300::/96

add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1

bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1

<!--NeedCopy-->
```

Simple large scale NAT64 configuration with an extended ACL6 rule for identifying subscribers:

```
add ns acl6 LSN-NAT64-ACL-2 ALLOW  - srcIPv6 = 2001:DB8:5002::20 - 2001:DB8:5002::200

apply acl6s
```

```

add lsn client LSN-NAT64-CLIENT-2

bind lsn client LSN-NAT64-CLIENT-2  - acl6name LSN-NAT64-ACL-2

add lsn pool LSN-NAT64-POOL-2

bind lsn pool LSN-NAT64-POOL-2 203.0.113.5-203.0.113.10

add lsn ip6profile LSN-NAT64-PROFILE-2 -type NAT64 -natprefix 2001:DB8:302::/96

add lsn group LSN-NAT64-GROUP-2 -clientname LSN-NAT64-CLIENT-2 -ip6profile LSN-NAT64-PROFILE-2

bind lsn group LSN-NAT64-GROUP-2 -poolname LSN-NAT64-POOL-2

<!--NeedCopy-->
```

Large scale NAT64 configuration with deterministic NAT resource allocation:

```
add lsn client LSN-NAT64-CLIENT-7

bind lsn client LSN-NAT64-CLIENT-7 -network6 2001:DB8:1002::7/128

add lsn pool LSN-NAT64-POOL-7 -nattype DETERMINISTIC

bind lsn pool LSN-NAT64-POOL-7 203.0.113.24-203.0.113.27

add lsn ip6profile LSN-NAT64-PROFILE-7 -type NAT64 -natprefix 2001:DB8:307::/96

add lsn group LSN-NAT64-GROUP-7 -clientname LSN-NAT64-CLIENT-7 -ip6profile LSN-NAT64-PROFILE-7 -nattype DETERMINISTIC -portblocksize 256

bind lsn group LSN-NAT64-GROUP-7 -poolname LSN-POOL-7

<!--NeedCopy-->
```

Configuring Application Layer Gateways for Large Scale NAT64

September 21, 2020

For some Application layer protocols, the IP addresses and protocol port numbers are also communicated in the packet payload. Application Layer Gateway for a protocol parses the packet's payload and does necessary changes to ensure that the protocol continues to work over large scale NAT64.

The NetScaler appliance supports ALG for the following protocols for large scale NAT64:

- FTP
- ICMP
- TFTP
- SIP
- RTSP

Application Layer Gateway for FTP, ICMP, and TFTP Protocols

September 21, 2020

You can enable or disable ALG for the FTP protocol for an large scale NAT64 configuration by enabling or disabling the FTP ALG option of the LSN group of the configuration.

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

ALG for the TFTP protocol is disabled by default. TFTP ALG is enabled automatically for an large scale NAT64 configuration when you bind a UDP LSN application profile, with endpoint-independent-mapping, endpoint-independent filtering, and destination port as 69 (well-known port for TFTP), to the LSN group.

Application Layer Gateway for SIP Protocol

September 21, 2020

Using Large Scale NAT64 with Session Initiation Protocol (SIP) is complicated, because SIP messages contain IP addresses in the SIP headers as well as in the SIP body. When LSN is used with SIP, the SIP headers contain information about the caller and the receiver, and the device translates this information to hide it from the outside network. The SIP body contains the Session Description Protocol (SDP) information, which includes IP addresses and port numbers for transmission of the media. SIP ALG for large scale NAT64 is compliant with RFC 3261, RFC 3581, RFC 4566, and RFC 4475.

Limitations of SIP ALG

SIP ALG for large scale NAT64 has the following limitations:

- Only SDP payload is supported.
- The following are not supported:
  - Multicast IP addresses
  - Encrypted SDP
  - SIP TLS
  - FQDN translation
  - SIP layer authentication
  - Traffic Domains
  - Admin partitions
  - NetScaler clusters
  - Multipart body
  - Line folding

Configuring SIP ALG

You need to configure the SIP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuration Large Scale NAT64. While configuring LSN, make sure that you:

- Set the following parameters while adding an LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Create a SIP ALG profile and make sure that you define either the source port range or destination port range. Bind the SIP ALG profile to the LSN group.
- Enable SIP ALG in the LSN group.

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-sipalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

To enable SIP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn sipalgprofile <sipalgprofilename>[-dataSessionIdleTimeout <positive_integer>][-sipSessionTimeout <positive_integer>] [-registrationTimeout <positive_integer>] [-sipsrcportrange <port[-port]>] [-sipdstportrange <port[-port]>] [-openRegisterPinhole ( ENABLED | DISABLED )] [-openContactPinhole ( ENABLED | DISABLED )] [-openViaPinhole ( ENABLED | DISABLED )] [-openRecordRoutePinhole ( ENABLED | DISABLED )]-sipTransportProtocol ( TCP | UDP ) [-openRoutePinhole ( ENABLED | DISABLED )] [-rport ( ENABLED | DISABLED )]

show lsn sipalgprofile <sipalgprofilename
<!--NeedCopy-->
```

Sample Configuration

The following sample large scale NAT64 configuration, SIP ALG is enabled for TCP traffic from subscriber devices in the network 2001:DB8:1003::/96.

```
add lsn client LSN-NAT64-CLIENT-9

Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96

Done
add lsn pool LSN-NAT64-POOL-9

Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90

Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8:309::/96

Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT

Done
add lsn sipalgprofile SIPALGPROFILE-9  -sipdstportrange 5060  -sipTransportProtocol TCP

```

```
Done
add lsn group LSN-NAT64-GROUP-9 -clientnameLSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -sipalg ENABLED

Done
bind lsn group LSN-NAT64-GROUP-9 -poolnameLSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
    PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9 -sipalgprofilename SIPALGPROFILE-9
Done
<!--NeedCopy-->
```

Application Layer Gateway for RTSP Protocol

September 21, 2020

Real Time Streaming Protocol (RTSP) is an application-level protocol for the transfer of real-time media data. Used for establishing and controlling media sessions between end points, RTSP is a control channel protocol between the media client and the media server. The typical communication is between a client and a streaming media server.

Streaming media from a private network to a public network requires translating IP addresses and port numbers over the network. NetScaler functionality includes an Application Layer Gateway (ALG) for RTSP, which can be used with Large Scale NAT (LSN) to parse the media stream and make any necessary changes to ensure that the protocol continues to work over the network.

How IP address translation is performed depends on the type and direction of the message, and the type of media supported by the client-server deployment. Messages are translated as follows:

- Outbound request—Private IP address to NetScaler owned public IP address called LSN IP address.
- Inbound response—LSN IP address to private IP address.
- Inbound request—No translation.
- Outbound response—Private IP address to LSN pool IP address.

Limitations of RTSP ALG

The RTSP ALG does not support the following:

- Multicast RTSP sessions

- RTSP session over UDP
- Admin partitions
- NetScaler clusters
- RTSP Authentication
- HTTP tunneling

Configuring RTSP ALG

Configure RTSP ALG as part of the LSN configuration. For instructions on configuring LSN, see Configuring Large Scale NAT64. While configuring, make sure that you:

- Set the following parameters while adding an LSN application profile:
  - IP Pooling = PAIRED
  - Address and Port Mapping = ENDPOINT-INDEPENDENT
  - Filtering = ENDPOINT-INDEPENDENT
- Enable RTSP ALG in the LSN group
- Create a RTSP ALG profile and bind the RTSP ALG profile to the LSN group

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-rtspalg ( ENABLED |
    DISABLED )]

show lsn group <groupname>
<!--NeedCopy-->
```

To enable RTSP ALG for an LSN configuration by using the CLI

At the command prompt, type:

```
add lsn rtspalgprofile <rtspalgprofilename> [-rtspIdleTimeout <
    positive_integer>] -rtspportrange <port[-port]> [-
    rtspTransportProtocol (TCP|UDP)]

show lsn rtspalgprofile <rtspalgprofilename>
<!--NeedCopy-->
```

Sample RTSP ALG Configuration

The following sample large scale NAT64 configuration, RTSP ALG is enabled for TCP traffic from subscriber devices in the network 2001:DB8:1002::/96.

```
add lsn client LSN-NAT64-CLIENT-9
Done
bind lsn client LSN-NAT64-CLIENT-9 -network6 2001:DB8:1002::/96
Done
add lsn pool LSN-NAT64-POOL-9
Done
bind lsn pool LSN-NAT64-POOL-9 203.0.113.90
Done
add lsn ip6profile LSN-NAT64-PROFILE-9 -type NAT64 -natprefix 2001:DB8
    :309::/96
Done
add lsn appsprofile LSN-NAT64-APPS-PROFILE-9 TCP -ippooling PAIRED -
    mapping ENDPOINT-INDEPENDENT -filtering ENDPOINT-INDEPENDENT
Done
add lsn rtspalgprofile RTSPALGPROFILE-9 -rtspIdleTimeout 1000 -
    rtspportrange 554
Done
add lsn group LSN-NAT64-GROUP-9 -clientname LSN-NAT64-CLIENT-9 -
    ip6profile LSN-NAT64-PROFILE-7 -rtspalg ENABLED
Done
bind lsn group LSN-NAT64-GROUP-9 -poolname LSN-NAT64-POOL-9
Done
bind lsn group LSN-NAT64-GROUP-9 -appsprofilename LSN-NAT64-APPS-
    PROFILE-9
Done
bind lsn group LSN-NAT64-GROUP-9  -rtspalgprofilename RTSPALGPROFILE-9
Done
<!--NeedCopy-->
```

Configuring Static Large Scale NAT64 Maps

September 21, 2020

The NetScaler appliance supports manual creation of NAT64 mappings, which contain the mapping between the following information:

- Subscriber’s IP address and port
- NAT IP address and port

Static Large Scale NAT64 mappings are useful in cases where you want to ensure that the IPv4 connections initiated to a NAT IP address:port are IPv6 translated and mapped to the subscriber IP address:port (for example, web servers located in the internal network).

To create a Large Scale NAT64 mapping by using the command line

At the command prompt, type:

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

show lsn static
<!--NeedCopy-->
```

Wildcard Port Static Large Scale NAT64 Maps

A static large scale NAT64 mapping entry is usually a one-to-one mapping between a subscriber IPv6 address:port and a NAT IPv4 address:port. A one-to-one static large scale NAT64 mapping entry exposes only one port of the subscriber IP address to the Internet.

Some situations might require exposing all ports (64K - limited to the maximum number of ports of a NAT IPv4 address) of a subscriber IP address to the Internet (for example, a server hosted on an internal network and running a different service on each port). To make these internal services accessible through the Internet, you have to expose all the ports of the server to the Internet.

One way to meet this requirement is to add 64 thousand one-to-one static mapping entries, one mapping entry for each port. Creating those entries is very cumbersome and a big task. Also, this large number of configuration entries might lead to performance issues in the NetScaler appliance.

A simpler method is to use wildcard ports in a static mapping entry. You just need to create one static mapping entry with NAT-port and subscriber-port parameters set to the wildcard character (*), and the protocol parameter set to ALL, to expose all the ports of a subscriber IP address for all protocols to the Internet.

For a subscriber's inbound or outbound connections matching a wildcard static mapping entry, the subscriber's port does not change after the NAT operation. When a subscriber-initiated connection to the Internet matches a wildcard static mapping entry, the NetScaler appliance assigns a NAT port that has the same number as the subscriber port from which the connection is initiated. Similarly, an Internet host gets connected to a subscriber's port by connecting to the NAT port that has the same number as the subscriber's port.

To configure the NetScaler appliance to provide access to all ports of a subscriber IPv6 address, create a wildcard static map with the following mandatory parameter settings:

- Protocol=ALL
- Subscriber port = *
- NAT port = *

In a wildcard static map, unlike in a one-to-one static map, setting the NAT IP parameter is mandatory. Also, the NAT IP address assigned to a wildcard static map cannot be used for any other subscribers.

To create a wildcard static map by using the command line interface

At the command prompt, type:

```
add lsn static <name> ALL <subscrIP> *  <natIP> * [-td <positive_integer>] [-destIP <ip_addr>

show lsn static
<!--NeedCopy-->
```

In the following sample configuration of a wildcard static map, all ports of a subscriber whose IP address is 2001:DB8:5001::3 are made accessible through NAT IP 203.0.11.33.

```
add lsn static NAT64-WILDCARD-STATIC-1 ALL 2001:DB8:5001::3 * 203.0.113.33 *
Done
<!--NeedCopy-->
```

Logging and Monitoring Large Scale NAT64

September 21, 2020

You can log large scale NAT64 information to diagnose and troubleshoot problems, and to meet legal requirements. You can monitor the performance of the large scale NAT64 deployment by using statistical counters and displaying the related current sessions.

Logging Large Scale NAT64

Logging large scale NAT64 information is required for ISPs to meet legal requirements and identify the source of traffic at any given time.

A log message for a large scale NAT64 mapping entry consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced.
- Time stamp.

- Entry type (MAPPING).
- Whether the mapping entry was created or deleted.
- Subscriber's IP address, port, and traffic domain ID.
- NAT IP address and port.
- Protocol name.
- Destination IP address, port, and traffic domain ID might be present, depending on the following conditions:
  - Destination IP address and port are not logged for endpoint-independent mapping.
  - Only the destination IP address is logged for address-dependent mapping. The port is not logged.
  - Destination IP address and port are logged for address-port-dependent mapping.

A log message for a large scale NAT64 session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp
- Entry type (SESSION)
- Whether the session is created or removed
- Subscriber's IP address, port, and traffic domain ID
- NAT IP address and port
- Protocol name
- Destination IP address, port, and traffic domain ID

The following table displays sample large scale NAT64 log entries of each type stored on the configured log servers. The log entries show that a subscriber whose IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

Log Entry Type	Sample Log Entry
Session Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Log Entry Type	Sample Log Entry
Mapping Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
Session Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Configuration Steps

You can configure logging of large scale NAT64 information for a large scale NAT64 configuration by setting the LSN groups's logging and session logging parameters. These are group level parameters and are disabled by default. The NetScaler appliance logs large scale NAT64 sessions for an LSN group only when both logging and session logging parameters are enabled.

The following table displays the logging behavior for an LSN group for various settings of logging and session logging parameters.

Logging	Session Logging	Logging Behavior
Enabled	Enabled	Logs LSN mapping entries as well as LSN sessions
Enabled	Disabled	Logs LSN mapping entries but not LSN sessions
Disabled	Enabled	Logs neither mapping entries nor LSN sessions

To log large scale NAT64 information by using the CLI

To set the logging and session logging parameters while adding an LSN group, at the command prompt, type:

```
add lsn group <groupname> -clientname <string> [-logging (ENABLED|
    DISABLED)] [-sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

To set the logging and session logging parameters for an existing LSN group, at the command prompt, type:

```
set lsn group <groupname> [-logging (ENABLED|DISABLED)] [-
    sessionLogging (ENABLED|DISABLED)]

show lsn group
<!--NeedCopy-->
```

Sample Configuration

In this example of large scale NAT64 configuration, logging and session logging paramters are enabled for LSN group LSN-NAT64-GROUP-1.

The NetScaler appliance logs large scale NAT64 session and mapping information for connections from subscribers (in the network 2001:DB8:5001::/96).

Sample configuration:

```
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1  -logging ENABLED -sessionLogging
    ENABLED
Done
```

```
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
<!--NeedCopy-->
```

Logging MSISDN Information for Large Scale NAT64

A Mobile Station Integrated Subscriber Directory Number (MSISDN) is a telephone number uniquely identifying a subscriber across multiple mobile networks. The MSISDN is associated with a country code and a national destination code identifying the subscriber’s operator.

You can configure a NetScaler appliance to include MSISDNs in large scale NAT64 LSN log entries for subscribers in mobile networks. The presence of MSISDNs in the LSN logs facilitates faster and accurate back tracing of a mobile subscriber who has violated a policy or law, or whose information is required by lawful interception agencies.

The following sample LSN log entries include MSISDN information for a connection from a mobile subscriber in an LSN configuration. The log entries show that a mobile subscriber whose MSISDN is E164:5556543210 and IPv6 address is 2001:db8:5001::9 was connected to destination IP:port 23.0.0.1:80 through the NAT IP:port 203.0.113.63:45195 on April 7, 2016, from 14:07:57 GMT to 14:10:59 GMT.

Log Entry Type	Sample Log Entry
Session Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_SESSION 5532 0 : SESSION CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP
Mapping Creation	04/07/2016:14:07:57 GMT Informational 0-PPE-10 : default LSN LSN_ADDR_MAPPING 5533 0 : ADM CREATED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:TD 23.0.0.1:80, Protocol: TCP
Session Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_SESSION 25012 0 : SESSION DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Log Entry Type	Sample Log Entry
Mapping Deletion	04/07/2016:14:10:59 GMT 0-PPE-10 : default LSN LSN_ADDR_MAPPING 25013 0 : ADM DELETED **E164:5556543210** Client IP-Port:TD **2001:db8:5001::9-34937:0**, NatIP:NatPort 203.0.113.63:45195, Destination IP:Port:TD 23.0.0.1:0:80, Protocol: TCP

Configuration Steps

Perform the following tasks for including MSISDN information in LSN logs:

- **Create an LSN log profile**. An LSN log profile includes the log subscriber ID parameter, which specifies whether to or not to include the MSISDN information in the LSN logs of an LSN configuration.
- Bind the LSN log profile to an LSN group of an LSN configuration.Bind the created LSN log profile to an LSN group of an LSN configuration by setting the log profile name parameter to the created LSN log profile name. MSISDN information is included in all LSN logs related to mobile subscribers of this LSN group.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename> -logSubscriberID ( ENABLED |
    DISABLED )

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an NAT64 LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname>  -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

Sample Configuration

In this example of NAT64 LSN configuration, the LSN log profile LOG-PROFILE-MSISDN-1 has the log subscriber ID parameter enabled. LOG-PROFILE-MSISDN-1 is bound to LSN group LSN-NAT64-GROUP-1. MSISDN information is included in the LSN session and LSN mapping logs for connections from mobile subscribers (in network 2001:DB8:5001::/96).

```
add lsn logprofile  LOG-PROFILE-MSISDN-1  -logSubscriberID ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8:300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -logprofilename  LOG-PROFILE-MSISDN-1
Done
<!--NeedCopy-->
```

Compact Logging for Large Scale NAT

Logging LSN information is one of the important functions needed by ISPs to meet legal requirements and be able to identify the source of traffic at any given time. This eventually results in a huge volume of log data, requiring the ISPs to make large investments to maintain the logging infrastructure.

Compact logging is a technique for reducing the log size by using a notational change involving short codes for event and protocol names. For example, C for client, SC for session created, and T for TCP. Compact logging results in an average of 40 percent reduction in log size.

Configuration Steps

Perform the following tasks for logging LSN information in compact format:

1. Create an LSN log profile. An LSN log profile includes the Log Compact parameter, which specifies whether to or not to log information in compact format for an LSN configuration.
2. Bind the LSN log profile to an LSN group of an LSN configuration. Bind the created LSN log profile to an LSN group of an LSN configuration by setting the Log Profile Name parameter to the created LSN log profile name. All sessions and mappings for this LSN group are logged in compact format.

To create an LSN log profile by using the CLI

At the command prompt, type:

```
add lsn logprofile <logprofilename> -logCompact (ENABLED|DISABLED)

show lsn logprofile
<!--NeedCopy-->
```

To bind an LSN log profile to an LSN group of an LSN configuration by using the CLI

At the command prompt, type:

```
bind lsn group <groupname> -logProfileName <lsnlogprofilename>

show lsn group
<!--NeedCopy-->
```

Sample Configuration for NAT64:

```
add lsn logprofile  LOG-PROFILE-COMPACT-1 -logCompact ENABLED
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
```

```
Done
bind lsn group LSN-NAT64-GROUP-1  - logProfileName LOG-PROFILE-COMPACT-1
Done
<!--NeedCopy-->
```

Logging HTTP Header Information

The NetScaler appliance can log request header information of an HTTP connection that is using the NetScaler large scale NAT64 functionality. The following header information of an HTTP request packet can be logged:

- URL that the HTTP request is destined to
- HTTP Method specified in the HTTP request
- HTTP version used in the HTTP request
- IPv6 address of the subscriber that sent the HTTP request

The HTTP header logs can be used by ISPs to see the trends related to the HTTP protocol among a set of subscribers. For example, an ISP can use this feature to find out the most popular website among a set of subscribers.

Configuration Steps

Perform the following tasks for configuring the NetScaler appliance to log HTTP header information:

- Create an HTTP header log profile. An HTTP header log profile is a collection of HTTP header attributes (for example, URL and HTTP method) that can be enabled or disabled for logging.
- Bind the HTTP header to an LSN group of a large scale NAT64 configuration. Bind the HTTP header log profile to an LSN group of an LSN configuration by setting the HTTP header log profile name parameter to the name of the created HTTP header log profile. The NetScaler appliance then logs HTTP header information of any HTTP requests related to the LSN group. An HTTP header log profile can be bound to multiple LSN groups, but an LSN group can have only one HTTP header log profile.

To create an HTTP header log profile by using the the command line interface

At the command prompt, type:

```
add lsn httphdrlogprofile <httphdrlogprofilename> [-logURL ( ENABLED |
    DISABLED )] [-logMethod ( ENABLED | DISABLED )] [-logVersion (
    ENABLED | DISABLED )] [-logHost ( ENABLED | DISABLED )]

show lsn httphdrlogprofile
<!--NeedCopy-->
```

To bind an HTTP header log profile to an LSN group by using the the command line interface

At the command prompt, type:

```
bind lsn group <groupname> -httphdrlogprofilename <string>

show lsn group <groupname>
<!--NeedCopy-->
```

Sample Configuration

```
add lsn httphdrlogprofile HTTP-HEADER-LOG-1
Done
add lsn client LSN-NAT64-CLIENT-1 Done
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8
    :300::/96
Done
add lsn group LSN-NAT64-GROUP-1 -clientname LSN-NAT64-CLIENT-1  -
    ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1 -httphdrlogprofilename HTTP-HEADER-LOG
    -1
Done
<!--NeedCopy-->
```

Displaying Current Large Scale NAT64 Sessions

You can display the current large scale NAT64 sessions in order to detect any unwanted or inefficient sessions on the NetScaler appliance. You can display all or some large scale NAT64 sessions on the basis of selection parameters.

> **Note**
>
> When more than a million large scale NAT64 sessions exist on the NetScaler appliance, Citrix rec-

> ommends using the selection parameters to display selected large scale NAT64 sessions instead of displaying all of them.

To display all large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
show lsn session -nattype NAT64
<!--NeedCopy-->
```

To display selective large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
show lsn session -nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
    <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

Displaying Large Scale NAT64 Statistics

You can display statistics related to large scale NAT64 module, and evaluate its performance or troubleshoot problems. You can display a summary of statistics of all large scale NAT64 configurations or of a particular large scale NAT64 configuration. The statistical counters reflect events since the NetScaler appliance was last restarted. All these counters are reset to 0 when the NetScaler appliance is restarted.

To display total statistics of large scale NAT64 by using the command line interface

At the command prompt, type:

```
stat lsn nat64
<!--NeedCopy-->
```

To display statistics for a specified large scale NAT64 configuration by using the command line interface

At the command prompt, type:

```
stat lsn group <groupname>
<!--NeedCopy-->
```

Clearing Large Scale NAT64 Sessions

You can remove any unwanted or inefficient large scale NAT64 sessions from the NetScaler appliance. The appliance immediately releases resources (such as NAT IP address, port, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected large scale NAT64 sessions from the NetScaler appliance.

To clear all large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
flush lsn session  - nattype NAT64

show lsn session  - nattype NAT64
<!--NeedCopy-->
```

To clear selective large scale NAT64 sessions by using the command line interface

At the command prompt, type:

```
flush lsn session  - nattype NAT64 [-network6 <ipv6_addr|*>] [-
    clientname <string>] [-natIP <ip_addr> [-natPort <port>]]

show lsn session  - nattype NAT64 [-network6 <ipv6_addr|*>] [-clientname
     <string>] [-natIP <ip_addr> [-natPort <port>]]
<!--NeedCopy-->
```

Sample configuration:

Clear all large scale NAT64 sessions existing on a NetScaler appliance

```
flush lsn session   - nattype NAT64
Done
<!--NeedCopy-->
```

Clear all large scale NAT64 sessions related to client entity LSN-NAT64-CLIENT-1

```
flush lsn session  - nattype NAT64 -clientname LSN-NAT64-CLIENT-1
Done
<!--NeedCopy-->
```

Clear all large scale NAT64 sessions related to a subscriber network (2001:DB8:5001::/96) of LSN client entity LSN-NAT64-CLIENT-2

```
flush lsn session -nattype NAT64 -network6 2001:DB8:5001::/96 -
   clientname LSN-NAT64-CLIENT-2
Done
<!--NeedCopy-->
```

Port Control Protocol for Large Scale NAT64

September 21, 2020

NetScaler appliances now support Port Control Protocol (PCP) for large scale NAT (LSN). Many of an ISP's subscriber applications must be accessible from Internet (for example, Internet of Things (IOT) devices, such as an IP camera that provides surveillance over the Internet). One way to meet this requirement is to create static large scale NAT (LSN) maps. But for a very large number of subscribers, creating static LSN NAT maps is not a feasible solution.

Port Control Protocol (PCP) enables a subscriber to request specific LSN NAT mappings for itself and/or for other 3rd party devices. The large scale NAT device creates an LSN map and sends it to the subscriber. The subscriber sends the remote devices on the Internet the NAT IP address:NAT port at which they can connect to the subscriber.

Applications usually send frequent keep-alive messages to the large scale NAT device so that their LSN mappings do not time out. PCP helps reduce the frequency of such keep-alive messages by enabling the applications to learn the timeout settings of the LSN mappings. This helps reduce bandwidth consumption on the ISP's access network and battery consumption on mobile devices.

PCP is a client-server model and runs over the UDP transport protocol. A NetScaler appliance implements the PCP server component and is compliant with RFC 6887.

Configuration Steps

Perform the following tasks for configuring PCP:

- **(Optional) Create a PCP profile**. A PCP profile includes settings for PCP related parameters (for example, to listen for mapping and peer PCP requests). A PCP profile can be bound to a PCP server. A PCP profile bound to a PCP server applies all its settings to the PCP server. A PCP profile can be bound to multiple PCP servers. By default, one PCP profile with default parameters settings is bound to all PCP servers. A PCP profile that you bind to a PCP server overrides the default PCP profile settings for that server. A default PCP profile has the following parameter settings:
  - Mapping: Enabled

  - Peer: Enabled
  - Minimum map life: 120 seconds
  - Maximum max life: 86400 seconds
  - Announce count: 10
  - Third Party: Disabled
- **Create a PCP server and bind a PCP profile to it**. Create a PCP server on the NetScaler appliance to listen for PCP related requests and messages from the subscribers. A Subnet IP (SNIP) or (SNIP6) address must be assigned to a PCP server to access it. By default, a PCP server listens on port 5351.
- **Bind the PCP server to an LSN group of an LSN configuration**. Bind the created PCP server to an LSN group of an LSN configuration by setting the PCP Server parameter to specify the created PCP server. The created PCP server can be accessed only by the subscribers of this LSN group.

> **Note**
>
> A PCP server for a large scale NAT configuration does not serve requests from subscribers that are identified from ACL rules.

To create a PCP profile by using the CLI

At the command prompt, type:

```
add pcp profile <name> [-mapping ( ENABLED | DISABLED )] [-peer (
    ENABLED | DISABLED )] [-minMapLife <secs>]  [-maxMapLife <secs>] [-
    announceMultiCount <positive_integer>][-thirdParty ( ENABLED |
    DISABLED )]

show pcp profile <name>
<!--NeedCopy-->
```

To create a PCP server by using the CLI

At the command prompt, type:

```
add pcp server <name> <IPAddress> [-port <portNum|*>] [-pcpProfile <
    string>]

show pcp server <name>
<!--NeedCopy-->
```

Sample Configuration for NAT64

In the following sample configuration, PCP server PCP-SERVER-1, with PCP settings from PCP-PROFILE-1,is bound to LSN group LSN-NAT64-GROUP-1. PCP-SERVER-1 serves PCP requests from IPv6 subscribers in network 2001:DB8:5001::/96.

Sample configuration:

```
add pcp profile PCP-PROFILE-1 -minMapLife 400
Done
add pcp server PCP-SERVER-1 2001:DB8:6001::90  -pcpProfile PCP-PROFILE-1
Done
add lsn client LSN-NAT64-CLIENT-1
Done
bind lsn client LSN-NAT64-CLIENT-1 -network6 2001:DB8:5001::/96
Done
add lsn pool LSN-NAT64-POOL-1
Done
bind lsn pool LSN-NAT64-POOL-1 203.0.113.61 - 203.0.113.70
Done
add lsn ip6profile LSN-NAT64-PROFILE-1 -type NAT64 -natprefix 2001:DB8:300::/96
Done
add lsn group LSN-NAT64-PROFILE-1 -clientname LSN-NAT64-CLIENT-1  -ip6profile LSN-NAT64-PROFILE-1
Done
bind lsn group LSN-NAT64-GROUP-1 -poolname LSN-NAT64-POOL-1
Done
bind lsn group LSN-NAT64-GROUP-1  - pcpServer PCP-NAT64-SERVER-1
Done
<!--NeedCopy-->
```

Telco subscriber management

September 21, 2020

The number of subscribers in a telco network is increasing at an unprecedented rate, and managing them is becoming a challenge for service providers. Newer, faster, and smarter devices are placing high demand on the network and the subscriber management systems. It is no longer feasible to provide each subscriber the same standard of service, and the need for traffic processing on a per-subscriber basis is imperative.

The NetScaler appliance provides the intelligence to profile subscribers on the basis of their information stored in the Policy and Charging Rules Function (PCRF). When a mobile subscriber connects to the Internet, the packet gateway associates an IP address with the subscriber and forwards the data packet to the appliance. The appliance receives the subscriber information dynamically, or you can configure static subscribers. This information enables the appliance to apply its rich traffic management capabilities, such as content switching, integrated caching, rewrite, and responder, on a per-subscriber basis to manage the traffic.

Before you configure the NetScaler appliance to manage subscribers, you must allocate memory to the module that stores subscriber sessions. For dynamic subscribers, you must configure an interface through which the appliance receives session information. Static subscribers must be assigned IDs, and you can associate them with policies.

You can also do the following:

- Subscriber policy enforcement and management.
- Configure the appliance to uniquely identify a subscriber by using only the IPv6 prefix instead of the complete IPv6 address.
- Use policies to optimize TCP traffic for both dynamic and static subscribers. These policies associate different TCP profiles with different types of users.
- Manage idle sessions on a NetScaler appliance.
- Enable logging to a log server.
- Remove LSN sessions for deleted subscriber sessions.

Allocating memory for the subscriber session store module

Each subscriber session entry consumes 1 KB of memory. Storing 500,000 subscriber sessions at any point in time requires 500 MB of memory. This value must be added to the minimum memory requirement, which is shown as part of the output of the “show extendedmemoryparam” command. In the following example, the output is for a NetScaler VPX instance with 3 packet engines and 8 GB memory.

To store 500,000 subscriber sessions on this appliance, the configured memory must be 2058+500 MB (500,000 x 1 KB = 500 MB.)

> **Note**
>
> The configured memory must be in multiples of 2 MB and must not exceed the maximum memory usage limit. The appliance must be restarted for the changes to take effect.

Example

```
show extendedmemoryparam
        Extended Memory Global Configuration. This memory is utilized by
      LSN and Subscriber Session Store Modules:
        Active Memory Usage: 0 MBytes
        Configured Memory Limit: 0 MBytes
        Minimum Memory Required: 2058 MBytes
        Maximum Memory Usage Limit: 2606 MBytes
    Done
    set extendedmemoryparam -memLimit 2558
    Done
    show extendedmemoryparam
            Extended Memory Global Configuration. This memory is utilized by LSN and Subscriber Session Store Modules:

            Active Memory Usage: 2558 MBytes
            Configured Memory Limit: 2558 MBytes
            Minimum Memory Required: 2058 MBytes
            Maximum Memory Usage Limit: 2606 MBytes
    Done
<!--NeedCopy-->
```

Configure an interface for dynamic subscribers

The NetScaler appliance dynamically receives the subscriber information through any of the following types of interface:

- Gx Interface
- RADIUS Interface
- RADIUS and Gx Interface

> **Note**
>
> - In an HA setup, the subscriber sessions are continually synchronized on the secondary node. In the event of a failover, the subscriber information is still available on the secondary node.

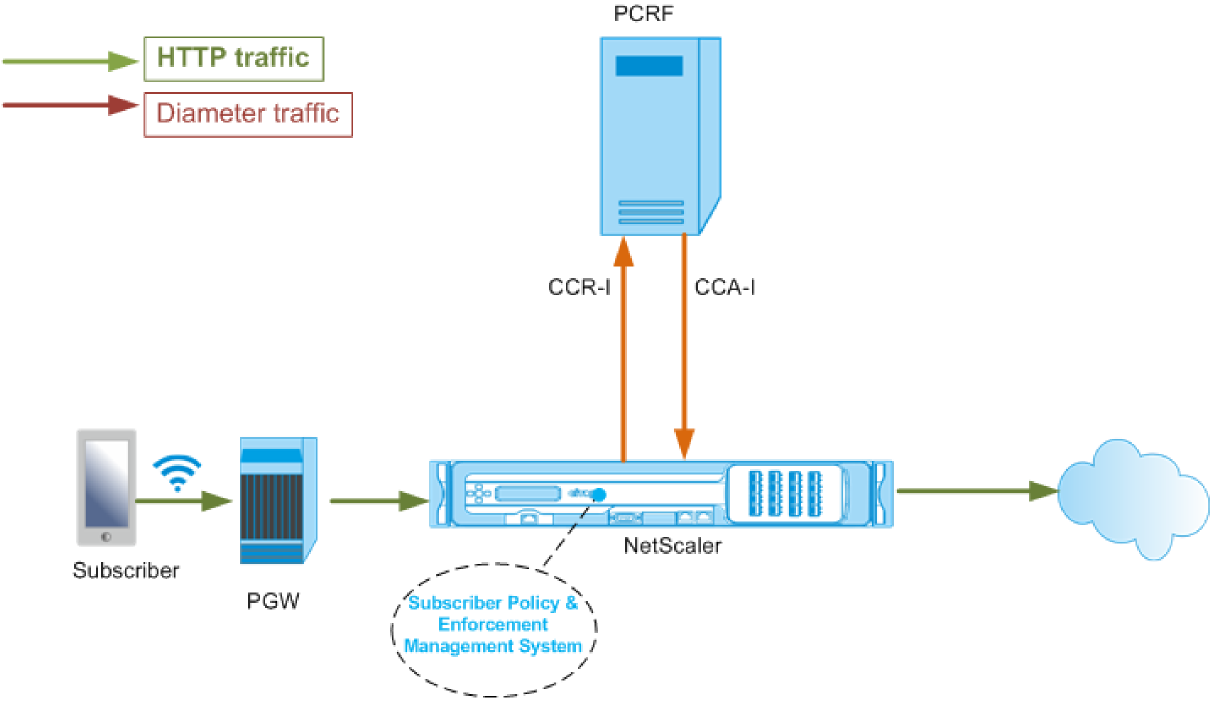
Gx interface

A Gx interface (as specified in 3GPP 29.212) is a standard interface based on the Diameter protocol that allows exchange of policy control and charging rules between a PCRF and a Policy and Charging Enforcement Function (PCEF) entity in a Telco network.

As soon as an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the PCRF as a Diameter message. When the data packet arrives at the appliance from packet gateway (PGW), the appliance uses the subscriber IP address to query the PCRF to get the subscriber information. This is also known as secondary PCEF functionality.

The Policy and Charging Control (PCC) rules received by the appliance over the Gx interface are stored on the appliance for the duration of the subscriber session, that is, until the PCRF sends a Re-Auth-Request (RAR) message with a Session-Release-Cause AVP or the subscriber session is terminated from the CLI or the configuration utility. If there are any updates to an existing subscriber, the PCRF sends the updates in an RAR message. A subscriber session is initiated when a subscriber logs on to the network, and terminated when the subscriber logs off.

The following illustration shows the high-level traffic flow. It assumes that the data plane traffic is HTTP. The appliance sends a Credit Control Request (CCR) over a Gx interface to the PCRF server and, in the credit control answer (CCA), receives the PCC rules and, optionally, other information, such as the Radio Access Technology (RAT) type, that applies to the particular subscriber. PCC rules include one or more policy (rule) names and other parameters. The appliance uses this information to retrieve the predefined rules stored on the appliance, and to direct the flow of traffic. It also stores this information in the subscriber policy and enforcement management system for the duration of the subscriber session. After a subscriber session is terminated, the appliance discards all the information about the subscriber.



The following example shows the commands for configuring a Gx interface. The commands are in boldface.

To set up a Gx interface, perform the following tasks

Add a DIAMETER service for each Gx interface. For example:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

Add a non-addressable DIAMETER load balancing virtual sever and bind the services created in step 1 to this virtual server. For more than one service, specify a persistenceType and the persistAVPno so that specific sessions are handled by the same PCRF server. For example:

```
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -persistAVPno 263

bind lb vserver vdiam pcrf-svc1

bind lb vserver vdiam pcrf-svc2
<!--NeedCopy-->
```

Configure NetScaler diameter identity and realm. Identity and realm are used as Origin-Host and Origin-Realm AVPs in diameter messages sent by the Gx client. For example:

```
set ns diameter  - identity netscaler.com  - realm com
<!--NeedCopy-->
```

Configure the Gx interface to use the virtual server created in step 2 as the PCRF virtual server. Specify the PCRF realm to use as Destination-Realm AVP in diameters messages sent by the Gx client. For example:

```
set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf.com
<!--NeedCopy-->
```

Set the subscriber interface type to GxOnly. For example:

```
set subscriber param -interfaceType GxOnly
<!--NeedCopy-->
```

To see the Gx interface configuration and status, type:

```
show subscriber gxinterface
<!--NeedCopy-->
```

Example

```
show subscriber gxinterface
    Gx Interface parameters:
        PCRF Vserver: vdiam (UP)
        Gx Client Identity...:  netscaler.com
        Gx Client Realm ..........:  com
        PCRF Realm: pcrf.com
        Hold Packets On Subscriber Absence: YES
        CCR Request Timeout: 10 Seconds
        CCR Request Retry Attempts: 3
        RevalidationTimeout: 1200 Seconds
        NegativeTTL: 120 Seconds
        ServicePath AVP code:262099 ServicePath AVP VendorID: 3845
        PCRF Connection State: Gx Connection Established with PCRF.
    Done

<!--NeedCopy-->
```

ARGUMENTS

vServer

Name of the load balancing or content switching virtual server to which the Gx connections are established. The service type of the virtual server must be DIAMETER or SSL_DIAMETER. This parameter is mutually exclusive with the service parameter. Therefore, you cannot set both service and the virtual server in the Gx interface.

Service

Name of DIAMETER or SSL_DIAMETER service corresponding to PCRF to which the Gx connection is established. This parameter is mutually exclusive with the vserver parameter. Therefore, you cannot set both service and the virtual server in the Gx Interface.

pcrfRealm

The realm of PCRF to which the message is to be routed. This is the realm used in Destination-Realm AVP by NetScaler Gx client (as a Diameter node).

holdOnSubscriberAbsence

Set to Yes to hold packets until the subscriber session information is fetched from the PCRF server. If set to No, the default subscriber profile is applied until the subscriber session information is fetched from the PCRF server. If a default subscriber profile is not configured, an UNDEF is raised for expressions that use subscriber attributes.

requestTimeout

Time, in seconds, within which the Gx CCR request must complete. If the request does not complete within this time, the request is retransmitted for the number of times specified in the requestRetryAttempts parameter. If request is not complete even after retransmitting, then the default subscriber profile is applied to this subscriber. If a default subscriber profile is not configured, an UNDEF is raised for expressions that use subscriber attributes. Zero disables the timeout. Default value: 10

requestRetryAttempts

Specify the number of times a request must be retransmitted if the request does not complete within the value specified in the requestTimeout parameter. Default value: 3

revalidationTimeout

Time, in seconds, after which the Gx CCR-U request is sent after any PCRF activity on a session. Any RAR or CCA message resets the timer. Zero value disables the idle timeout.

negativeTTL

Time, in seconds, after which the Gx CCR-I request is resent for sessions that have not been resolved by PCRF because the server is down or there is no response or a failed response is received. Instead of polling the PCRF server constantly, a negative-TTL makes the appliance hold on to an unresolved session. For negative sessions, the appliance inherits the attributes from the default subscriber profile, if one is configured and from the RADIUS accounting message, if one is received. Zero value disables the negative sessions. The appliance does not install negative sessions even if a subscriber session could not be fetched. Default value: 600

servicePathAVP

The AVP code in which PCRF sends the service path applicable to a subscriber.

servicePathVendorid

The vendor id of the AVP in which PCRF sends the service path applicable to a subscriber.

To configure Gx interface by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In Interface Type, select **GxOnly**.
4. Specify the values for the all required parameters.

5. Click **OK**.

RADIUS interface

With a RADIUS interface, the packet gateway forwards the subscriber information in a RADIUS Accounting Start message to the appliance through the RADIUS interface as soon as an IP-CAN session is established. A service of type RADIUSListener processes RADIUS Accounting messages. Add a shared secret for the RADIUS client. If a shared secret is not configured, the RADIUS message is silently dropped. The following example shows the commands for configuring a RADIUS interface. The commands are in boldface.

To set up a RADIUS interface, perform the following tasks:

Create a RADIUS listener service at the SNIP address where the RADIUS messages are received. For example:

```
add service srad1 192.0.0.206 RADIUSLISTENER 1813
<!--NeedCopy-->
```

Configure the subscriber RADIUS interface to use this service. For example:

```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

Set the subscriber interface type to RadiusOnly. For example:

```
set subscriber param -interfaceType RadiusOnly
<!--NeedCopy-->
```

Add a RADIUS client specifying a subnet and shared secret. For example:

```
add radius client 192.0.2.0/24 -radkey client123
<!--NeedCopy-->
```

A subnet of 0.0.0.0/0 implies that it is the default shared secret for all clients.To see the RADIUS interface configuration and status, type:

```
show subscriber radiusInterface
<!--NeedCopy-->
```

RADIUS Interface parameters:

Radius Listener Service: srad1(UP)

Done

Example:

```
add service pcrf-svc1 203.0.113.1 DIAMETER 3868

add service pcrf-svc2 203.0.113.2 DIAMETER 3868
<!--NeedCopy-->
```

ARGUMENTS

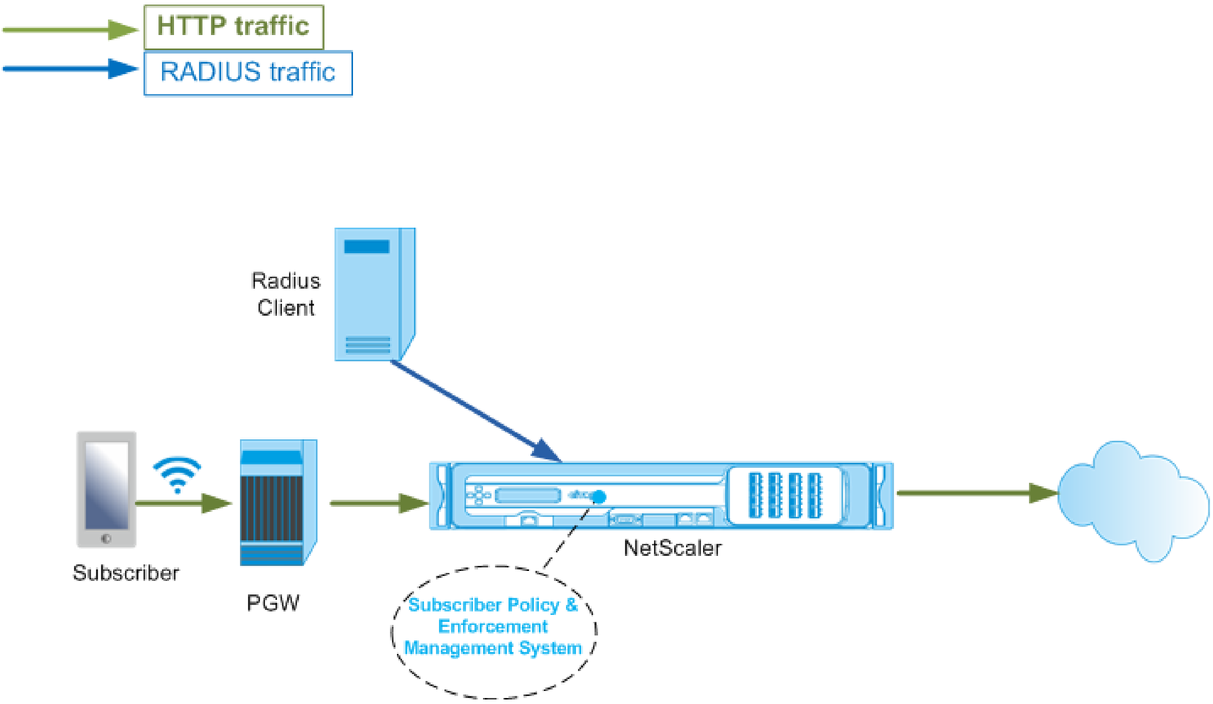
ListeningService

Name of the RADIUS listening service that will process the RADIUS accounting requests.

svrState

The state of the RADIUS listening service.

The following illustration shows the high-level traffic flow.


HTTP traffic
RADIUS traffic
Radius Client
Subscriber
PGW
NetScaler
Subscriber Policy & Enforcement Management System

To configure RadiusOnly interface by using the GUI

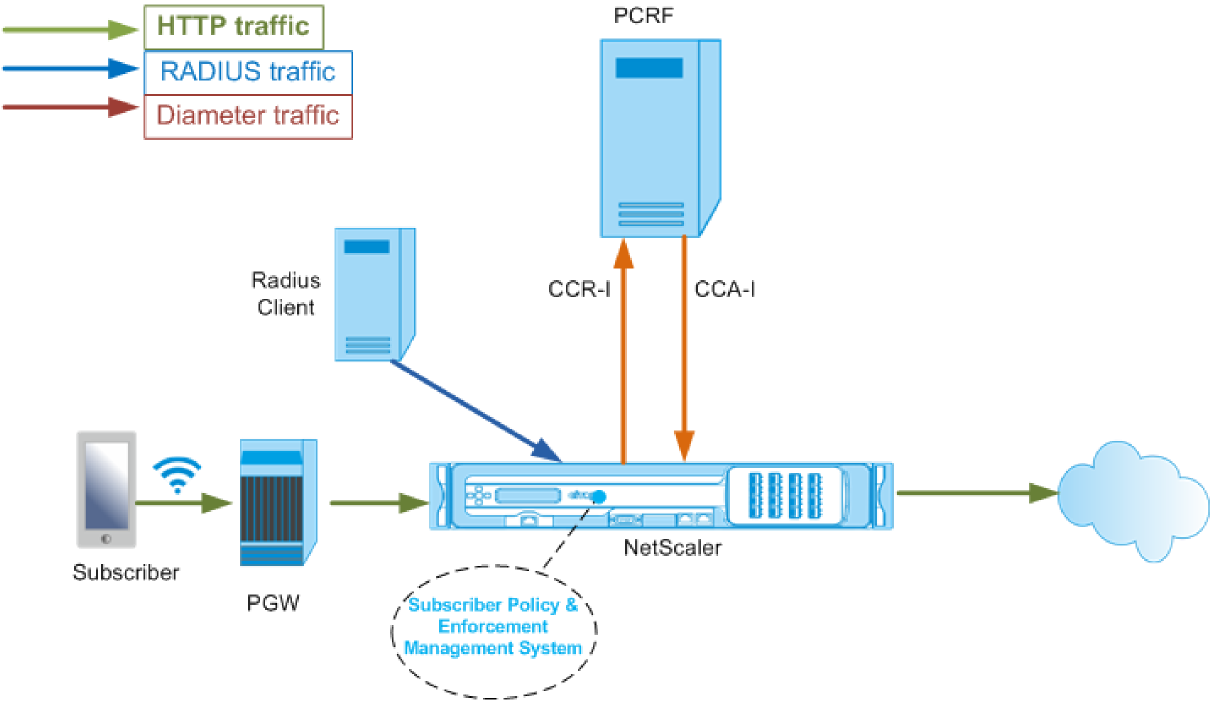
1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.
3. In Interface Type, select **RadiusOnly**.
4. Specify the values for the all required parameters.
5. Click **OK**.

RADIUS and Gx interface

With a RADIUS and Gx interface, as soon as an IP-CAN session is established, the packet gateway forwards the subscriber ID, such as the MSISDN, and Framed-IP address information about the subscriber to the appliance through the RADIUS interface. The appliance uses this subscriber ID to query the PCRF on the Gx interface to get the subscriber information. This is known as primary PCEF functionality. The following example shows the commands for configuring a RADIUS and Gx interface.

```
set subscriber param -interfaceType RadiusandGx
add service pcrf-svc 203.0.113.1 DIAMETER 3868
add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263
bind lb vserver vdiam pcrf-svc
set subscriber gxInterface -vServer vdiam -pcrfRealm testrealm1.net -
    holdOnSubscriberAbsence YES -revalidationTimeout 60 -negativeTTL 120
add service srad1 192.0.0.206 RADIUSLISTENER 1813    set subscriber
    radiusInterface -listeningService srad1
<!--NeedCopy-->
```

The following illustration shows the high-level traffic flow.


HTTP traffic
RADIUS traffic
Diameter traffic
PCRF
Radius Client
CCR-I
CCA-I
Subscriber
PGW
NetScaler
Subscriber Policy & Enforcement Management System

To configure RadiusAndGx interface by using the GUI

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. Click **Configure Subscriber Parameters**.

3. In Interface Type, select **RadiusAndGx**.
4. Specify the values for the all required parameters.
5. Click **OK**.

Configure static subscribers

You can configure the subscribers manually on the NetScaler appliance by using the command line or the configuration utility. You create static subscribers by assigning a unique subscriber ID and optionally associating a policy with each subscriber. The following examples show the commands for configuring a static subscriber.

In the following examples, **subscriptionIdvalue** specifies the international telephone number, and **subscriptionIdType** (E164 in this example) specifies the general format for international telephone numbers.

```
add subscriber profile 203.0.113.6 -subscriberRules policy1 policy2 -
    subscriptionIdType E164  - subscriptionIdvalue 98767543211
add subscriber profile 2002::a66:e8d3/64 -subscriberRules policy1
    policy3 -subscriptionIdtype E164  - subscriptionIdvalue 98767543212
```

To view the configured subscriber profiles, type:

> show subscriber profile

```
> show subscriber profile
1)      Subscriber IP: 2002::/64
        Profile Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
2)      Subscriber IP: 203.0.113.6
        Profile Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211
 Done
```

Default subscriber profile

A default subscriber profile is used if the subscriber IP address is not found in the subscriber session store on the appliance. In the following example, a default subscriber profile is added with the subscriber rule policy1.

```
> add subscriber profile * -subscriberRules policy1
```

View and clear subscriber sessions

Use the following command to display all the static and dynamic subscriber sessions.

> show subscriber sessions

```
> show subscriber sessions
1)      Subscriber IP: 2002::/64
        Session Attributes:
                Active Rules: policy1, policy3
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543212
2)      Subscriber IP: *
        Session Attributes:
                Active Rules: policy1
3)      Subscriber IP: 203.0.113.6
        Session Attributes:
                Active Rules: policy1, policy2
                Subscriber Id Type: E164
                Subscriber Id Value: 98767543211
4)      Subscriber IP: 192.168.0.11
        Session Attributes:
                Idle TTL remaining: 361 Seconds
                Active Rules: policy1
                Subscriber Id Type: E164
                Subscriber Id Value: 1234567811
                Service Path: policy1
                AVP(44): 34 44 32 42 42 38 41 43  2D 30 30 30 30 30 30
    31 31
                AVP(257): 00 01 C0 A8 0A 02
                PCRF-Host: host.pcrf.com
                AVP(280): 74 65 73 74 2E 63 6F 6D
…
Done
```

Use the following command to clear a single session or the complete session store. If you do not specify an IP address, the complete subscriber session store is cleared.

```
clear subscriber sessions <ip>
<!--NeedCopy-->
```

Subscriber policy enforcement & management system

The NetScaler appliance uses the subscriber's IP address as the key to the subscriber policy enforcement and management system.

You can add subscriber expressions to read the subscriber information available in the Subscriber Policy Enforcement & Management System. These expressions can be used with policy rules and actions that are configured for NetScaler features, such as integrated caching, rewrite, responder, and content switching.

The following commands are an example of adding a subscriber-based responder action and policy. The policy evaluates to true if the subscriber rule value is"pol1".

```
add responder action error_msg respondwith '"HTTP/1.1 403 OKrnrn" + "
    You are  not authorized to access Internet"'
add responder policy no_internet_access "SUBSCRIBER.RULE_ACTIVE("pol1")
    " error_msg
```

The following example shows the commands to add a subscriber-based rewrite action and policy. The action inserts an HTTP header "X-Nokia-MSISDN" by using the value of AVP(45) in the subscriber session.

```
> add rewrite action AddHDR-act insert_http_header X-Nokia-MSISDN "
    SUBSCRIBER.AVP(45).VALUE"
> add rewrite policy AddHDR-pol "HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)
    .EQUALS_ANY("patset-test")" AddHDR-act
```

In the following example, two policies are configured on the appliance. When the appliance checks the subscriber information and the subscriber rule is cache_enable, it performs caching. If the subscriber rule is cache_disable, the appliance does not perform caching.

```
> add cache policy nocachepol -rule "SUBSCRIBER.RULE_ACTIVE("
    cache_disable")" - action NOCACHE
> add cache policy cachepol -rule "SUBSCRIBER.RULE_ACTIVE("cache_enable
    ")" - action CACHE -storeInGroup cg1
```

For a complete list of expressions starting with "SUBSCRIBER." see the Policy Configuration Guide.

IPv6 prefix based subscriber sessions

A telco user is generally identified by the IPv6 prefix rather than the complete IPv6 address. The NetScaler appliance now uses the prefix instead of the complete IPv6 address (/128) to identify a subscriber in the database (subscriber store). For communicating with the PCRF server (for example, in a CCR-I message), the appliance now uses the framed-IPv6-Prefix AVP instead of the complete IPv6 address. The default prefix length is /64, but you can configure the appliance to use a different value.

To configure the IPv6 prefix by using the command line

set subscriber param [-ipv6PrefixLookupList ...]

The first example command below sets a single prefix and the second example command sets multiple prefixes.

```
set subscriber param -ipv6PrefixLookupList 64
set subscriber param -ipv6PrefixLookupList 64 72 96
```

To configure the IPv6 prefix by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and in **IPv6 Prefix Lookup List**, specify one or more prefixes.

Idle session management of subscriber sessions in a Telco network

Subscriber session cleanup on a NetScaler appliance is based on control plane events, such as a RADIUS Accounting Stop message, a Diameter RAR (session release) message, or a "clear subscriber session" command. In some deployments, the messages from a RADIUS client or a PCRF server might not reach the appliance. Additionally, during heavy traffic, the messages might be lost. A subscriber session that is idle for a long time continues to consume memory and IP resources on the NetScaler appliance. The idle session management feature provides configurable timers to identify idle sessions, and cleans up these sessions on the basis of the specified action.

A session is considered idle if no traffic from this subscriber is received on the data plane or the control plane. You can specify an update, terminate (inform PCRF and then delete the session), or delete (without informing PCRF) action. The action is taken only after the session is idle for the time specified in the idle timeout parameter.

To configure the idle session timeout and the associated action by using the command line

```
set subscriber param [-idleTTL <positive_integer>] [-idleAction <
    idleAction>]
<!--NeedCopy-->
```

Examples:

```
set subscriber param -idleTTL 3600 -idleAction ccrTerminate

set subscriber param -idleTTL 3600 -idleAction ccrUpdate

set subscriber param -idleTTL 3600 -idleAction delete
<!--NeedCopy-->
```

To disable the idle session timeout, set the idle timeout to zero.

set subscriber param –idleTTL 0

To configure the idle session timeout and the associated action by using the configuration utility

1. Navigate to **Traffic Management** > **Subscriber** > **Parameters**.
2. In the details pane, under **Settings**, click **Configure Subscriber Parameters** and specify an **Idle Time** and **Idle Action**.

Subscriber session event logging

If you enable subscriber logging, you can track the RADIUS and Gx control plane messages specific to a subscriber, and use the historical data to analyze subscriber activities. Some of the key attributes are MSISDN and time stamp. The following attributes are also logged:

- Session Event (Install, Update, Delete, Error)
- Gx Message Type (CCR-I, CCR-U, CCR-T, RAR)
- Radius Message Type (Start, Stop)
- Subscriber IP
- SubscriberID Type (MSISDN(E164), IMSI)
- SubscriberID value

By using these logs, you can track users by IP address and, if available, MSISDN.

You can enable subscriber session logging to a local or remote syslog or nslog server. The following example shows how to enable subscriber logging to a remote syslog server.

```
> add syslogAction sysact1 192.0.2.0 -loglevel EMERGENCY ALERT CRITICAL
    ERROR WARNING NOTICE INFORMATIONAL -subscriberlog enabled
```

From these logs, you can learn about any activity related to a user, such as the time when a session was updated, deleted, or created (installed). Additionally, error messages are also logged.

Examples:

1. The following log entries are examples of RADIUSandGx session creation, session update, and session deletion.

```
09/30/2015:16:29:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 147 0 : Session Install, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001

09/30/2015:16:30:18 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 148 0 : Session Update, GX MsgType: CCR-U, IP: 100.10.1.1, ID: E164 - 30000000001

09/30/2015:17:27:56 GMT Informational 0-PPE-0 : default SUBSCRIBER SESSION_EVENT 185 0 : Session Delete, GX MsgType: CCR-T, RADIUS MsgType: Stop, IP: 100.10.1.1, ID: E164 - 30000000001
```

2. The following log entries are examples of failure messages, such as when a subscriber is not found on the PCRF server and when the appliance cannot connect to the PCRF server.

```
09/30/2015:16:44:15 GMT Error 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 169 0 : Failure Reason: PCRF failure response, GX MsgType: CCR-I, IP: 100.10.1.1

Sep 30 13:03:01 09/30/2015:16:49:08 GMT 0-PPE-0 : default SUBSCRIBER SESSION_FAILURE 176 0 : Failure Reason: Unable to connect to PCRF, GX MsgType: CCR-I, RADIUS MsgType: Start, IP: 100.10.1.1, ID: E164 - 30000000001#000#000#000#000#000#000#000#000#000#00
```

Subscriber aware LSN session termination

In earlier releases, if a subscriber session is deleted when a RADIUS Accounting STOP or a PCRF-RAR message is received, or as a result of any other event, such as TTL expiry or flush, the corresponding LSN sessions of the subscriber are removed only after the configured LSN timeout period. LSN sessions that are kept open until this timeout expires continue to consume resources on the appliance.

From release 11.1, a new parameter (subscrSessionRemoval) is added. If this parameter is enabled, and the subscriber information is deleted from the subscriber database, LSN sessions corresponding to that subscriber are also removed. If this parameter is disabled, the subscriber sessions are timed out as specified by the LSN timeout settings.

To configure subscriber aware LSN session termination by using the CLI

At the command prompt, type:

```
set lsn parameter -subscrSessionRemoval (          DISABLED )
ENABLED
```

```
> set lsn parameter -subscrSessionRemoval ENABLED
Done
> sh lsn parameter
```

```
            LSN Global Configuration:

            Active Memory Usage: 0 MBytes
            Configured Memory Limit: 0 MBytes
            Maximum Memory Usage Limit: 912 MBytes
            Session synchronization: ENABLED
            Subscriber aware session removal: ENABLED
```

To configure subscriber aware LSN session termination by using the GUI

1. Navigate to **System** > **Large Scale NAT**.
2. In **Getting started**, click **Set LSN Parameter**.
3. Set the **Subscriber Aware Session Removal parameter**.

Troubleshooting

If your deployment is not working as expected, use the following commands to troubleshoot:

- show subscriber gxinterface
  This command's output can include the following error messages (shown here with suggested responses):
  - Gx Interface Not Configured-Use set subscriber param command to configure the correct interface type.
  - PCRF not configured-Configure a Diameter vServer or Service on GxInterface-Use the set subscriber gx interface command to assign a Diameter virtual server or service to this interface.
  - PCRF is not ready-Check corresponding vserver/service for more details-Use the show LB vserver or show service command to check the state of the service.
  - NetScaler is waiting for CEA from PCRF-Capability negotiation between the PCRF and NetScaler might be failing. This could be an intermittent state. If it persists, check the DIAMETER settings on your PCRF server.
  - Memory is not configured to store subscriber sessions. Please use 'set extendedmemoryparam -memlimit <>'-Use the set extendedmemoryparam command to configure extended memory.
- show subscriber radiusinterface
  If "Not Configured" is the output of this command, use the set subscriber radiusinterface command to specify a RADIUSListener service.

If subscriber logging is enabled, you can get more detailed information from the log files.

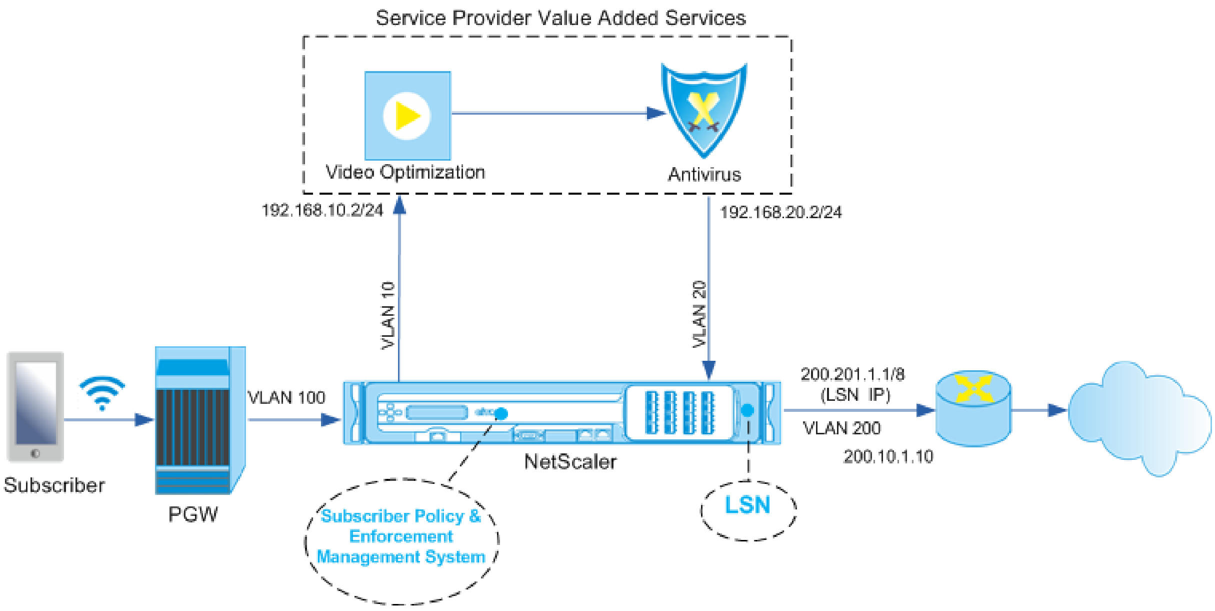
Subscriber aware traffic steering

September 21, 2020

Traffic steering directs subscriber traffic from one point to another. When a subscriber connects to the network, the packet gateway associates an IP address with the subscriber and forwards the data packet to the NetScaler appliance. The appliance communicates with the PCRF server over the Gx interface to get the policy information. Depending on the policy information, the appliance performs one of the following actions:

- Forward the data packet to another set of services (as shown in the following illustration).
- Drop the packet.
- Perform only Large Scale NAT (LSN), if LSN is configured on the appliance.

The values shown in the following figure are configured in the CLI procedure that follows the figure. A content switching virtual server on the NetScaler appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.


Service Provider Value Added Services
Video Optimization
Antivirus
192.168.10.2/24
192.168.20.2/24
VLAN 10
VLAN 20
VLAN 100
Subscriber
PGW
NetScaler
Subscriber Policy & Enforcement Management System
LSN
200.201.1.1/8 (LSN IP)
VLAN 200
200.10.1.10

To configure traffic steering for the above deployment by using the CLI

Add the appliance's subnet IP (SNIP) addresses.

Example:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 100.100.100.1 255.0.0.0 -type snip

add ns ip 200.200.200.1 255.0.0.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses.

Example:

```
add vlan 10

add vlan 20

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.1.1.1 255.0.0.0
<!--NeedCopy-->
```

Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

Example:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
   servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

Example:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

The service path AVP that contains this name must already be configured as part of the global configuration. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

Example:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
```

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

Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

The appliance performs LSN by default. To override LSN, you must create a net profile with the overrideLsn parameter enabled, and bind this profile to all the load balancing virtual servers that are configured for value added services (VASs).

Example:

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

Configure the VAS on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vint sint
```

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

Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define expressions that associate a virtual server with a service function.

Example:

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule  SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
    SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

To configure traffic steering on the appliance by using the GUI

1. Navigate to **System**> **Network**> **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management**> **Service Chaining**> **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management**> **Subscriber**> **Parameters**> **Configure Subscriber Parameters** and specify the following:
    - Interface Type: Specify **RadiusAndGx**.
    - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
    - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management**> **Service Chaining**> **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System**> **Network**> **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System**> **Network**> **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System**> **Network**> **Settings**> **Configure Layer 3 Parameters** and verify

that **Override LSN** is not selected.

8. Navigate to **Traffic Management**> **Load Balancing**> **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management**> **Content Switching**> **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

To configure service chaining on the appliance by using the GUI

1. Navigate to **System**> **Network**> **IPs** and add the subnet IP addresses.
2. Navigate to System > Network > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management**> **Service Chaining**> **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management**> **Subscriber**> **Parameters**> **Configure Subscriber Parameters** and specify the following:
    - Interface Type: Specify **RadiusAndGx**.
    - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
    - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management**> **Service Chaining**> **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System**> **Network**> **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System**> **Network**> **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System**> **Network**> **Settings**> **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management**> **Load Balancing**> **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management**> **Content Switching**> **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

Subscriber aware service chaining

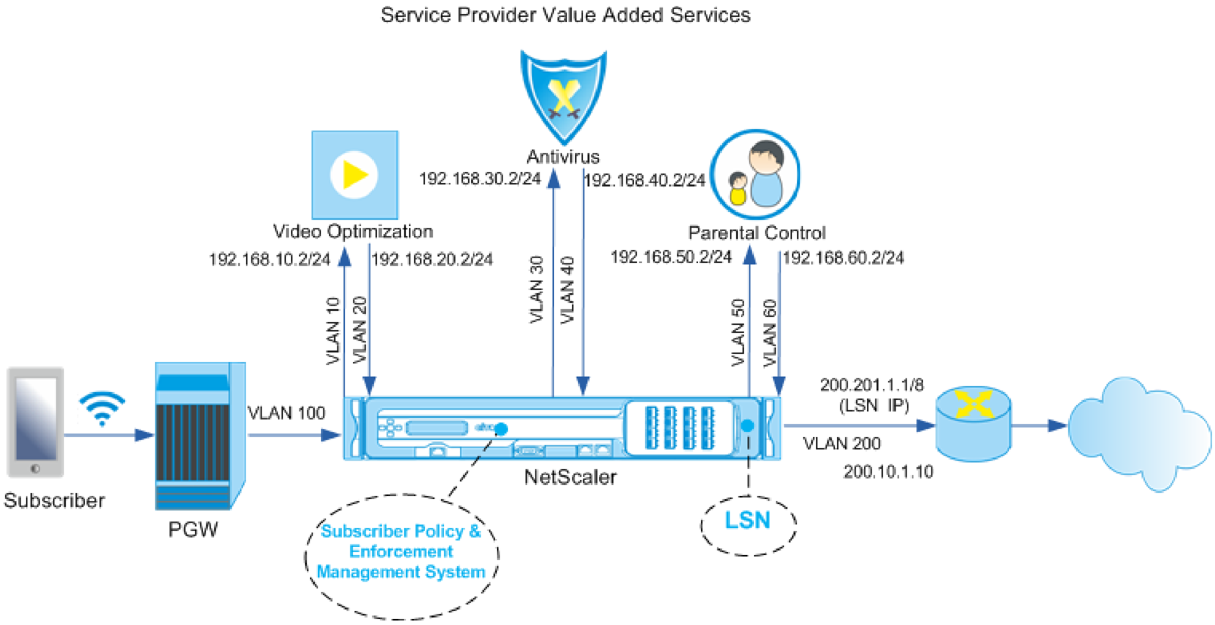
September 21, 2020

With the huge increase in the data traffic passing through telco networks, it is no longer feasible for service providers to steer all the traffic through all the value added services (VAS). A service provider should be able to optimize usage of VAS and intelligently steer traffic to improve the user experience. For example, video optimization is not required for traffic that does not include a video. Moreover, if a subscriber is connected to a 4G network, content can be streamed in high definition (HD), and video optimization might not be needed. However, video optimization improves the experience for a user in a 3G network. Similarly, caching provides a faster and better user experience and can be enabled depending on the subscriber plan. Another example of VAS is parental control. If parents provide a mobile handset to a minor child, they would like some kind of control over the websites that their child visits.

To do the above and more, service providers must be able to provide value-added services on a per-subscriber basis. In other words, entities in the service provider network must be capable of extracting the subscriber information and intelligently steering the packet on the basis of this information.

Service chaining determines the set of services through which the traffic from a subscriber must pass before going to the Internet. Instead of sending all the traffic to all the services, the NetScaler intelligently routes all requests from a subscriber to a specific set of services on the basis of the policy defined for that subscriber.

The following figure shows the entities involved in service chaining. The values shown are configured in the procedure that follows the figure. A content switching virtual server on the NetScaler appliance directs requests to the value added services or skips them, depending on the defined rule, and then sends the packet out to the Internet after performing LSN.


Service Provider Value Added Services
Antivirus
192.168.30.2/24
192.168.40.2/24
Video Optimization
192.168.10.2/24
192.168.20.2/24
Parental Control
192.168.50.2/24
192.168.60.2/24
VLAN 10
VLAN 20
VLAN 30
VLAN 40
VLAN 50
VLAN 60
VLAN 100
200.201.1.1/8
(LSN IP)
VLAN 200
200.10.1.10
Subscriber
PGW
NetScaler
Subscriber Policy &
Enforcement
Management System
LSN

To configure service chaining for the above deployment by using the CLI

Add the appliance's subnet IP (SNIP) addresses.

Example:

```
add ns ip 192.168.10.1 255.255.255.0 -type snip

add ns ip 192.168.20.1 255.255.255.0 -type snip

add ns ip 192.168.30.1 255.255.255.0 -type snip

add ns ip 192.168.40.1 255.255.255.0 -type snip

add ns ip 192.168.50.1 255.255.255.0 -type snip

add ns ip 192.168.60.1 255.255.255.0 -type snip

add ns ip 100.1.1.1 255.0.0.0 -type snip

add ns ip 200.201.1.1 255.0.0.0 -type snip
<!--NeedCopy-->
```

Add the VLANs. VLANs help the appliance identify the source of the traffic. Bind the VLANs to the interfaces and subnet IP addresses. Add an ingress and an egress VLAN for each VAS.

Example:

```
add vlan 10

add vlan 20

add vlan 30

add vlan 40

add vlan 50

add vlan 60

add vlan 100

add vlan 200

bind vlan 10 -ifnum 1/4 -tagged -IPAddress 192.168.10.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.20.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.30.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.40.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.50.1 255.255.255.0

bind vlan 20 -ifnum 1/4 -tagged -IPAddress 192.168.60.1 255.255.255.0

bind vlan 100 -ifnum 1/2 -tagged -IPAddress 100.1.1.1 255.0.0.0

bind vlan 200 -ifnum 1/3 -tagged -IPAddress 200.201.1.1 255.0.0.0
<!--NeedCopy-->
```

Specify the VLAN on which the subscriber traffic arrives on the appliance. Specify the service path AVP that tells the appliance where to look for the service path name within the subscriber session. For primary PCEF functionality, specify the interfaceType as RadiusAndGx.

Example:

```
set ns param -servicePathIngressVLAN 100

set subscriber gxinterface -servicepathAVP 1001 1005 -
    servicepathVendorid 10415

set subscriber param -interfaceType RadiusAndGx
<!--NeedCopy-->
```

Configure a service and virtual server of type Diameter, and bind the service to the virtual server. Then, specify the PCRF realm and subscriber Gx interface parameters. For primary PCEF functionality, configure a RADIUS listener service and RADIUS interface.

Example:

```
add service sd1 10.102.232.200 DIAMETER 3868

add lb vserver vdiam DIAMETER 0.0.0.0 0 -persistenceType DIAMETER -
    persistAVPno 263

bind lb vserver vdiam sd1

set ns diameter -identity netscaler.sc1.net -realm pcrf1.net

set subscriber gxInterface -vServer vdiam -pcrfRealm pcrf1.net -
    holdOnSubscriberAbsence YES -idleTTL 1200 -negativeTTL 120

add service srad1 10.102.232.236 RADIUSListener 1813

```

```
set subscriber radiusInterface -listeningService srad1
<!--NeedCopy-->
```

Add service functions to associate a VAS with an ingress VLAN. Add a service path to define the chain, that is, specify the VAS that the packet must be sent to and the order in which it must go to that VAS. The service path name is usually sent by the PCRF. However, the service path of the default subscriber profile (*) applies if any of the following is true:

- PCRF does not have the subscriber information.
- The subscriber information does not include this AVP.
- The appliance is unable to query the PCRF. For example, the service representing the PCRF is DOWN.

The service path AVP that contains this name must be configured as part of the global configuration earlier. Bind the service function to the service path. The service index specifies the order in which the VAS is added to the chain. The highest number (255) indicates the beginning of the chain.

Example:

```
add ns servicefunction SF1 -ingressVLAN 20

add ns servicefunction SF2 -ingressVLAN 40

add ns servicefunction SF3 -ingressVLAN 60

add ns servicepath pol1

bind ns servicepath pol1 -servicefunction SF1 -index 255

bind ns servicepath pol1 -servicefunction SF2 -index 254

bind ns servicepath pol1 -servicefunction SF3 -index 253

add ns servicepath pol2

bind ns servicepath pol2 -servicefunction SF2 -index 255

add ns servicepath pol3

bind ns servicepath pol3 -servicefunction SF1 -index 255

add subscriber profile * -subscriberrules default_path
<!--NeedCopy-->
```

Add the LSN configuration. That is, define the NAT pool and identify the clients for which the appliance must perform LSN.

Example:

```
add lsn pool pool1

bind lsn pool pool1 200.201.1.1

add lsn client client1

bind lsn client client1 -network 100.0.0.0 -netmask 255.0.0.0

add lsn group group1 -clientname client1

bind lsn group group1 -poolname pool1
<!--NeedCopy-->
```

The appliance performs LSN by default. To override LSN, you must create a net profile with overrideLsn parameter enabled and bind this profile to all the load balancing virtual servers that are configured for value added services (VASs).

Example:

```
add netprofile np1

set netprofile np1 -overrideLsn ENABLED

set lb vserver vs1 -netprofile np1
<!--NeedCopy-->
```

Configure the VAS on the appliance. This includes creating the services and virtual servers and then binding the services to the virtual servers.

Example:

```
add service vas1 192.168.10.2 ANY 80 -usip YES

add service vas2 192.168.30.2 ANY 80 -usip YES

add service vas3 192.168.50.2 ANY 80 -usip YES

add service sint 200.10.1.10 ANY 80 -usip YES

add lb vserver vs1 ANY -m MAC -l2Conn ON

add lb vserver vs2 ANY -m MAC -l2Conn ON

add lb vserver vs3 ANY -m MAC -l2Conn ON

add lb vserver vint ANY -m MAC -l2Conn ON

bind lb vserver vs1 vas1

bind lb vserver vs2 vas2

bind lb vserver vs3 vas3

bind lb vserver vint sint
<!--NeedCopy-->
```

Add the content switching (CS) configuration. This includes virtual servers, policies, and their associated actions. The traffic arrives at the CS virtual server and is then redirected to the appropriate load balancing virtual server. Define expressions that associate a virtual server with a service function.

Example:

```
add cs vserver cs1 ANY * 80 -l2Conn ON

add cs action csact1 -targetLBVserver vs1

add cs action csact2 -targetLBVserver vs2

add cs action csact3 -targetLBVserver vs3

add cs action csactint -targetLBVserver vint

add cs policy cspol1 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF1") &&
   SYS.VSERVER("vs1").STATE.EQ(UP)" -action csact1

add cs policy cspol2 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF2") &&
   SYS.VSERVER("vs2").STATE.EQ(UP)" -action csact2

add cs policy cspol3 -rule "SUBSCRIBER.SERVICEPATH.IS_NEXT("SF3") &&
   SYS.VSERVER("vs3").STATE.EQ(UP)" -action csact3

bind cs vserver cs1 -policyName cspol1 -priority 110

bind cs vserver cs1 -policyName cspol2 -priority 120

bind cs vserver cs1 -policyName cspol3 -priority 130

bind cs vserver cs1 -lbvserver vint
<!--NeedCopy-->
```

To configure service chaining on the appliance by using the GUI

1. Navigate to **System** > **Network** > **IPs** and add the subnet IP addresses.
2. Navigate to **System** > **Network** > **VLANs** and add VLANs, Bind the VLANs to the interfaces and subnet IP addresses.
3. Navigate to **Traffic Management** > **Service Chaining**> **Configure Service Path Ingress VLAN** and specify an ingress VLAN.
4. Navigate to **Traffic Management** > **Subscriber** > **Parameters** > **Configure Subscriber Parameters** and specify the following:
    - Interface Type: Specify **RadiusAndGx**.
    - Configure a diameter virtual server, PCRF realm, and the subscriber GX interface parameters.
    - Specify the RADIUS interface parameters.
5. Navigate to **Traffic Management** > **Service Chaining**> **Service Function** and add service functions to associate a value-added service with an ingress VLAN.
6. Navigate to **System**> **Network**> **Large Scale NAT**. Click **Pools** and add a pool. Click **Clients** and add a client. Click **Groups** and add a group and specify the client. Edit the group and bind the pool to this group.
7. Navigate to **System**> **Network**> **Net Profiles** and add a net profile. Select **Override LSN**. Optionally, navigate to **System**> **Network**> **Settings**> **Configure Layer 3 Parameters** and verify that **Override LSN** is not selected.
8. Navigate to **Traffic Management**> **Load Balancing**> **Virtual Servers** and configure the virtual servers and value-added services on the appliance. Bind the services and the net profile to the virtual server.
9. Navigate to **Traffic Management**> **Content Switching**> **Virtual Servers** and configure a virtual server, policy, and action. Specify the target load balancing virtual server.

Policy based TCP profile selection

September 21, 2020

You can configure the NetScaler appliance to perform TCP optimization based on subscriber attributes. For example, the appliance can select different TCP profiles at run time, based on the network to which the user equipment (UE) is connected. As a result, you can improve a mobile user's experience by setting some parameters in the TCP profiles and then using policies to select the appropriate profile.

Create separate TCP profiles for subscribers connecting through a 4G network and for users connecting through any other network. Define a policy rule that is selected on the basis of a subscriber parameter, such as RAT-type. In the following examples, if RAT-Type is EUTRAN, a TCP profile that supports a faster connection is selected (Example 1). For all other RAT-Type values, a different TCP profile is selected (Example 2).

> **Note**
>
> The RAT-Type AVP (AVP code 1032) is of type Enumerated and is used to identify the radio access technology that is serving the UE.
>
> The value "1004" indicates that the RAT is EUTRAN. (RFC 29.212).

Example1:

```
add ns tcpProfile tcp2 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd 16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 1000000 -flavor BIC - dynamicReceiveBuffering DISABLED -sendBuffsize 1000000 -dsack DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 500 -maxburst 15

add appqoe action appact2 -priority HIGH -tcpprofile tcp2

add appqoe policy apppol2 -rule "SUBSCRIBER.AVP(1032).VALUE.GET_UNSIGNED32(0, BIG_ENDIAN).EQ(1004)" -action appact2

bind cs vserver <name>  -policyname apppol2 -priority 20 -type request
<!--NeedCopy-->
```

Example2:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -initialCwnd 16 - oooQSize 15000 -slowStartIncr 1 -bufferSize 150000 -flavor BIC - dynamicReceiveBuffering DISABLED -sendBuffsize 150000 -dsack DISABLED -maxcwnd 4000000 -fack ENABLED -minRTO 200 -maxburst 15

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "SUBSCRIBER.AVP(1032).VALUE.GET_UNSIGNED32(0, BIG_ENDIAN).NE(1004)" -action appact1

bind cs vserver <name>  -policyname apppol1 -priority 10 -type request
<!--NeedCopy-->
```

Load Balance Control-Plane Traffic that is based on Diameter, SIP, and SMPP Protocols

September 21, 2020

With the increase in control-plane traffic, the servers can become a bottleneck because the traffic is not optimally distributed among the servers. Therefore, messages must be load balanced. The NetScaler appliance supports Diameter, SIP, and SMPP load balancing.

SIP

NetScaler enables you to load balance SIP messages over UDP or over TCP (including TLS) to a group of proxy servers. NetScaler also provides Call-ID based persistence and Call-ID hash load balancing method using which you direct packets for a particular SIP session to the same load balanced SIP server.
The NetScaler default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for SIP protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

For more information, see Load Balancing a Group of SIP Servers.

SMPP

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers.
The NetScaler appliance provides optimal distribution of messages across your servers, preventing poor performance and outages. The NetScaler appliance:

- Load balances messages originating from the server and from the client
- Monitors the health of the message centers
- Provides content switching support for the message centers
- Handles concatenated messages

Limitation: Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the NetScaler appliance responds with an error message.

For more information, see SMPP Load Balancing

Diameter

Diameter is a base protocol with more than 50 protocols (also called applications) built over it. Therefore, the diameter traffic generated in a Telco network is high. To optimally maintain this diameter traffic, the NetScaler appliance performs load balancing, content switching, and acts as a relay agent. Additionally, the appliance offers rewrite and responder functionality. The appliance supports rate limiting of Diameter messages.

For more information, see Configuring Diameter Load Balancing.

Provide DNS Infrastructure/Traffic Services, such as, Load Balancing, Caching, and Logging for Telecom Service Providers

September 21, 2020

Telecom service providers can configure the NetScaler appliance to function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the NetScaler appliance. This enables the NetScaler appliance to provide quick responses for repeated translations and hence enhances the customer experience and also saves the bandwidth. The caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the NetScaler appliance returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler appliance. The NetScaler appliance then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler appliance serves the Address record of the domain from the cache without querying the configured DNS server and hence saves the bandwidth.

From 11.0 release onwards, NetScaler also logs the DNS requests that it receives and also the responses that it sends to the client. Telecom service providers can use this log to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshooting

For more information, see
Domain Name System.

Provide Subscriber Load Distribution Using GSLB Across Core-Networks of a Telecom Service Provider

September 21, 2020

Scalability, high availability and performance are critical to service provider deployments. While many service providers deploy there infrastructure at a single location or multiple location, these deployments are subject to a number of inherent limitations, such as:

- If the site loses connectivity to all or part of the public Internet, it will be inaccessible to users and customers, which can have significant impact on the business.
- Users accessing the site from geographically distant locations may experience large and highly variable delays, which are exacerbated by the large number of round trips that HTTP requires to transfer content.

NetScaler appliance's Global Server Load Balancing (GSLB) overcomes these problems by distributing traffic among sites deployed in multiple geographic locations. By serving content from many different points in the Internet, GSLB alleviates the impact of network bandwidth bottlenecks and provides robustness in case of network failures at a particular site. Users can be automatically directed to the nearest or least loaded site at the time of the request, minimizing the likelihood of long download delays and/or service disruptions.
You can use NetScaler appliance's global server load balancing for:

- Disaster recovery or high availability by configuring an Active-standby data center setup that consists of an active and a standby data center. When a failover occurs as a result of a disaster event, the standby data center becomes operational.
- High availability and speed by configuring an active-active data center setup that consists of multiple active data centers. Client requests are load balanced across active data centers.
- Directing client requests to the data center that is closest in geographical distance or network distance by configuring a proximity setup.
- Full-DNS resolutions, GSLB processes DNS queries of the A, AAAA and CNAME types, and the DNS function option can process DNS queries of all other types, such as MX and PTR. Also, if the recursive resolution is enabled, the appliance will forward DNS queries for domain names that are not configured on the NetScaler appliance.

For more information, see Global Server Load Balancing.

Bandwidth Utilization Using Cache Redirection Functionality

September 21, 2020

The volume of web traffic on the Internet is enormous and a large percentage of that traffic is redundant. Multiple clients ask web servers for the same content repeatedly leading to inefficient use of bandwidth. To relieve the origin web server of processing each request, Internet Service Providers (ISPs) can use the cache redirection feature of NetScaler appliance and serve the content from a cache server instead of from the origin server. The NetScaler appliance analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers. Cache redirection feature of NetScaler is policy-based, and by default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

For more information, see Cache Redirection.

NetScaler TCP Optimization

September 21, 2020

The NetScaler appliance provides advanced TCP tuning and optimization techniques and capabilities that are well suited to modern 3.5 and 4G networks, improving user experience and perceived download speeds significantly.

This section focuses on detailed instructions relevant to:

- Choosing and inserting an appropriate NetScaler T1000 Series model in a mobile network for TCP optimization
- Full configuration instructions related to not only TCP optimization but also for appropriate Layer-2 and Layer-3 configuration of the T1 device

Getting Started

September 21, 2020

Hardware

Citrix provides a wide breath of NetScaler models that might be loosely based on two factors:

- Capacity, currently ranging from hundreds of Mbps for the low-end VPX appliance to 160Gbps for the high-end 25000 MPX series appliance
- Telco grade, with the availability of the T1000 series for Telco datacenters.

Your Citrix Sales or Support representative can help you select the appropriate hardware for your demo, trial, or production needs.

The remainder of this section uses a NetScaler T1200 as a reference hardware. Note that putting aside superficial differences related to number and notation of available interfaces (see ⋆ in note) or well-documented limitations of NetScaler VPX (see ⋆⋆ in note) the instructions should apply mostly verbatim regardless of the NetScaler model selected.

> **Note**
>
> ⋆ For instance the T1010 model only has 12x1GbE typically marked as 1/1-1/12 rather than the 10/x notation used in this document.
>
> ⋆⋆ A NetScaler VPX instance typically doesn't support LACP aggregation; it might also not support VLAN tagging.

Initial Setup

Through Serial Console

After a serial cable is connected, you can log on to the NetScaler appliance with the following credentials:

- Username: nsroot
- Password: nsroot

Once logged in, configure the basic details of the NetScaler appliance as shown in the screen capture below.

Example:

```
set ns config - IPAddress <ip_addr> -netmask <netmask>

saveconfig

reboot -warm
<!--NeedCopy-->
```

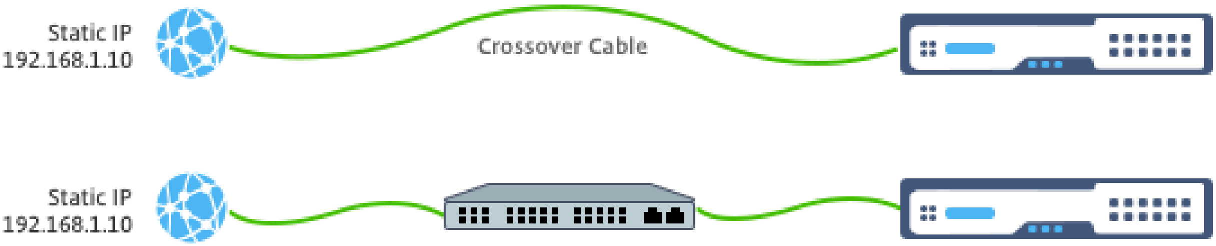
After you restart the appliance, you might use SSH for further configuration of the T1100 nodes.

Through LOM

Lights out Management (LOM) port on the front panel of the NetScaler appliance allows operator to remotely monitor and manage the appliance independently of the operating system. Operator can change the IP address, power cycle, and perform a code dump by connecting to the NetScaler appliance through the LOM port.

Default IP Address of LOM port is 192.168.1.3

Figure. Intial Configuration of LOM Module



Set a static IP on your laptop and plug it directly into the LOM interface with a crossover cable or into a switch in the same broadcast domain as the LOM interface.

For initial configuration, type the port's default address: http://192.168.1.3 in a web browser and change the LOM port's default IP address.

Refer to Configuration Guides for further details.

Software

The NetScaler TCP optimization for mobile networks is constantly evolving. The capabilities and tunings outlined in this document require a NetScaler Telco build. Here is an example showing the NetScaler Telco build.

Example:

```
show ver

NetScaler NS11.0: Build 64.957.nc, Date: Aug 26 2016, 02:00:23
<!--NeedCopy-->
```

If the T1000 has not shipped with the appropriate build revision, contact the NetScaler Customer Support.

> **Important**
>
> Both the appliances should have the same software image.

SSH Client

A NetScaler appliance can be configured by using either the CLI or the HTML5 GUI. However, this section provides only CLI-based instructions.

While the CLI is accessible through the NetScaler serial console, an SSH client is normally recommended to allow for remote NetScaler configuration.

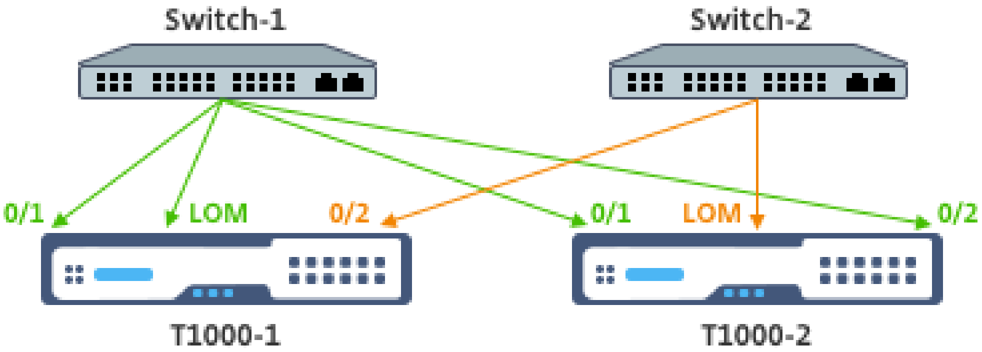
Management Network

September 21, 2020

Connectivity

Most NetScaler devices offer redundant 1GbE OAM ports, notated as 0/1 and 0/2. To provide for redundancy in case of a switch failure, you should connect the relevant ports to different upstream switches.

A high-level overview of the recommended connectivity is outlined in the following diagram:



After the NetScaler appliance is connected to the management network, subsequent configuration steps can be performed remotely using SSH or web connectivity to the CLI and GUI respectively.

Routing

The add route command may be used to configure any routes appropriate to the management network. The relevant gateway should be reachable on the NSIP subnet, as shown below.

Example:

```
add route <network>  <netmask>  <gateway>
<!--NeedCopy-->
```

Licensing

September 21, 2020

A valid license file should be installed on the NetScaler appliance. The license should support at least as many Gbps as the expected maximum Gi-LAN throughput.

License files should be copied through an SCP client to the /nsconfig/license of the appliance, as shown in the screen capture below.

EXample:

```
shell ls /nsconfig/license/

CNS_V3000_SERVER_PLT_Retail.lic ssl
<!--NeedCopy-->
```

Do a warm restart to apply the new license, as shown in the screen capture below.

EXample:

```
reboot -warm

Are you sure you want to restart NetScaler (Y/N)? [N]:y

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

After the restart completes, verify that the license has been properly applied, by using the show license CLI.

In the example below a 3Gbps Platinum license has been successfully installed.

EXample:

```
> show license

             License status:

                                     Web Logging: YES

                                               ...

                                 Model Number ID: 3000

                                    License Type: Platinum License

Done
```

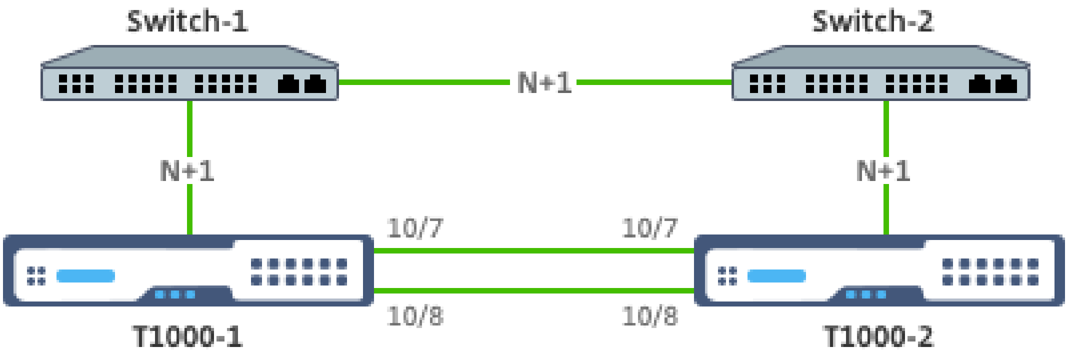
High Availability

September 21, 2020

High availability (HA) refers to an active-standby operational mode of a NetScaler device pair. Each device has its own dedicated management IP address. All other IP addresses are owned by the active device in the pair.

Connectivity

While there are multiple connectivity options for aNetScaler HA pair, the most recommended one is depicted in the following diagram:


Switch-1
Switch-2
N+1
N+1
N+1
10/7
10/7
10/8
10/8
T1000-1
T1000-2

In the above diagram, the N+1 red links between each T1000 and the respective switch imply N+1 redundancy - as explained in Connectivity. For instance, considering a 45 Gbps Gi-LAN N=5 is an appropriate value, with 6x10GbE LACP channels between each switch and the respective T1000 as well as between the two switches.

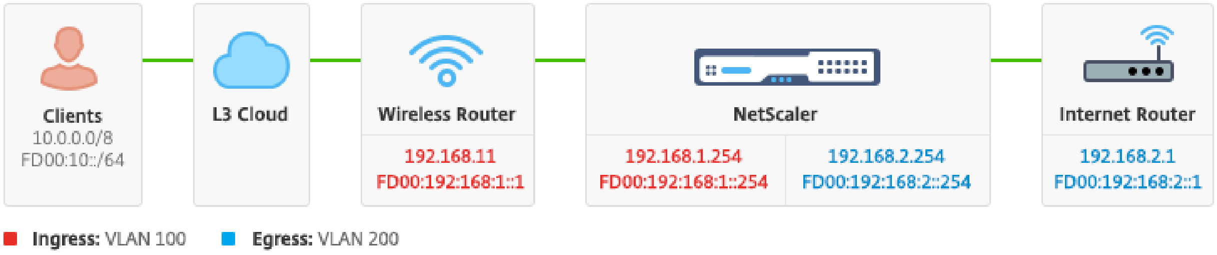
An extra pair of links is recommended between the NetScaler pair, to provide for HA communication isolation from the OAM network.

Gi-LAN Integration

September 21, 2020

Typically, a NetScaler appliance is inserted as a separate L3 inline node in the Gi-LAN, similarly to an L3 router.

Figure: A simple depiction of a Gi-LAN


Clients
10.0.0.0/8
FD00:10::/64
L3 Cloud
Wireless Router
192.168.11
FD00:192:168:1::1
NetScaler
192.168.1.254
FD00:192:168:1::254
192.168.2.254
FD00:192:168:2::254
Internet Router
192.168.2.1
FD00:192:168:2::1
Ingress: VLAN 100
Egress: VLAN 200

Connectivity

A physical NetScaler connectivity to upstream switches is recommended to provide for sufficient redundancy. For example, assuming that a NetScaler appliance is inserted in a Gi-LAN that is handling a total (uplink+downlink) of 24Gbps, connectivity with 4x10GbE or more interfaces is recommended. This effectively provides for N+1 redundancy in case of a link failure.

The relevant ports on the upstream switch should be configured for LACP port aggregation. The relevant configuration on NetScaler is outlined below:

Connectivity Configuration:

```
set interface 10/1 - tagall ON - lacpMode ACTIVE - lacpKey 1

set interface 10/2 - tagall ON - lacpMode ACTIVE - lacpKey 1

set interface 10/3 - tagall ON - lacpMode ACTIVE - lacpKey 1

set interface 10/4 - tagall ON - lacpMode ACTIVE - lacpKey 1
<!--NeedCopy-->
```

You can verify the appropriate functionality of LACP using the “show interface” command:

show interface:

sh interface LA/1

```
    1)      Interface LA/1 (802.3ad Link Aggregate) #39

            flags=0x4100c020 <ENABLED, UP, AGGREGATE, UP, HAMON, 802.1q>

            MTU=1500, native vlan=1, MAC=02:e0:ed:33:88:b0, uptime 340h11m56s

            Requested: media NONE, speed AUTO, duplex NONE, fctl NONE,

            throughput 0

            Actual: throughput 4000

            LLDP Mode: NONE,

            RX: Pkts(918446) Bytes(110087414) Errs(0) Drops(795989) Stalls(0)

```

```
              TX: Pkts(124113) Bytes(15255532) Errs(0) Drops(0) Stalls
         (0)

              NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0)
         Muted(0)

              Bandwidth thresholds are not set.

<!--NeedCopy-->
```

Disable the remaining unused interfaces and turn off the monitor.

set interface 10/5 –haMonitor OFF

Command:

```
set interface 10/24  - haMonitor OFF

disable interface 10/5

disable interface 10/24
<!--NeedCopy-->
```

Configuration of physical interfaces is not shared across the two NetScaler units. Hence, the above commands must be run across both NetScaler nodes in case of an HA pair deployment.

HA Configuration

All other configuration parameters are shared between the NetScaler nodes of an HA pair. Hence, HA sync should be enabled prior to any other configuration commands being run. Basic HA configuration involves the following steps:

1. Using the exact same NetScaler hardware, software, and license: HA pairs are not supported between different models (i.e. a T1100 and an MPX21550) or same models with different firmware levels. Refer to the appropriate instructions on upgrading an existing HA pair - Upgrading to Release 11.1.

2. Establishing the HA pair.

Example:

```
netscaler-1> add HA node 1 <netscaler-2-NSIP>

netscaler-2> add HA node 1 <netscaler-1-NSIP>
<!--NeedCopy-->
```

3. Verify the HA pair establishment running the following command in either node; both nodes should be visible, one of them as Primary (active), the other as a Secondary (standby).

Example:

"'show HA node

```
4\. Enable failsafe mode and maxFlips. This ensures that in case of a
   route monitor failure on both nodes at least one node remains active
    without active/standby status constantly switching.

**Example:**
```

set HA node –failsafe ON

set HA node -maxFlips 3 -maxFlipTime 1200

```
5\. Finally, enable HA sync to occur over the dedicated intra-NetScaler
    ports rather than the OAM network.

**Example:**
```

add vlan 4080 -aliasName syncVlan

set HA node -syncvlan 4080

```
> **Note**
>
> The VLAN 4080 in the commands in the above example shouldn' t be
   taken literally. Any unused VLAN-ID might be reserved.

## VLAN Configuration

After the physical interfaces have been appropriately configured, you
   might configure the appropriate Gi-LAN VLANs. For instance, consider
    a rather simple Gi-LAN environment with an ingress/egress VLAN pair
    with 100/101 VLAN-identifier respectively.

The following commands configure the relevant VLANs on top of the LACP
   channel created in the prior step.
```

add vlan 100
add vlan 101
bind vlan 100 –ifnum LA/1 –tagged
bind vlan 101 –ifnum LA/1 –tagged

```
## IPv4 Configuration
```

```

Typically, a NetScaler appliance requires one SNIP per VLAN. The example below assumes that the networks outlined in the Gi-LAN integration diagram, given in the begining of this page, have a /24 subnet mask:
```

add ns ip 192.168.1.254 255.255.255.0 –vserver DISABLED –mgmtAccess DISABLED
add ns ip 192.168.2.254 255.255.255.0 –vserver DISABLED –mgmtAccess DISABLED

```
After the SNIPs have been configured they should be associated with the appropriate VLAN:
```

bind vlan 100 –IPAddress 192.168.1.254 255.255.255.0
bind vlan 101 –IPAddress 192.168.2.254 255.255.255.0

```
## IPv4 Static Routing

The example outlined in the [Management Network](/en-us/netscaler/11-1/netscaler-support-for-telecom-service-providers/NS_TCP_optimization/NS_TCP_opt_mgmt_network.html) section calls for only a couple of static routing rules:

-  A 10.0.0.0/8 static route to the clients through the ingress router
-  A default route to the internet through the egress router

**Example:**
```

add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1

```
## IPv4 Policy-Based (VLAN - VLAN) routing

A NetScaler appliance allows for policy-based routing instead of static routing, with routing decisions usually keyed against the incoming interface and/or VLAN rather than destination IP. Policy-based routing is either a convenient alternative, in case the client source IP address range is subject to periodic changes, or a mandatory consideration, in case a packet’ s destination IP address is not sufficient by itself to reach a routing decision (i.e. in case of overlapping client IP addresses across multiple VLANs).

**Example:**
```

add ns pbr fromWirelessToInternet ALLOW –nextHop 192.168.2.1 –vlan 100 –priority 10

Done

add ns pbr fromInternetToWireless ALLOW –nextHop 192.168.1.1 –vlan 200 –priority 20

Done

apply ns pbrs

```
## IPv6 Configuration

The following commands assign IPv6 SNIP per vlan. The example below
    assumes that the networks outlined in the Figure: A simple depiction
     of a Gi-LAN in this page have a /64 subnet mask:

**Command:**
```

add ns ip6 fd00:192:168:1::254/64 -vServer DISABLED –mgmtAccess DISABLED
add ns ip6 fd00:192:168:2::254/64 -vServer DISABLED –mgmtAccess DISABLED
bind vlan 100 -IPAddress fd00:192:168:1::254/64
bind vlan 200 -IPAddress fd00:192:168:2::254/64

```
## IPv6 Routing

After  IPv6 addressing is complete, IPv6 static routing might be
    configured:

-  A fd00:10::/64 static route to the clients via the ingress router
-  A default route to the internet via the egress router

**Example:**
```

add route6 fd00:10::/64 fd00:192:168:1::1
add route6 ::/0 fd00:192:168:2::1

```
Or using policy-based routing:

**Example:**
```

add ns pbr6 fromWirelessToInternetv6 ALLOW -vlan 100 -priority 10 -nextHop fd00:192:168:2::1

add ns pbr6 fromInternetToWirelessv6 ALLOW -vlan 200 -priority 20 -nextHop fd00:192:168:1::1

apply ns pbr6

```
## LACP Redundancy and Failover

```

```
In case of an HA configuration, it’ s recommended to leverage the throughput option to configure a low threshold for the LACP channel. For instance, consider a 25Gbps Gi-LAN and a 4x10GbE channel between each NetScaler appliance in the HA pair and the upstream switch to provide N+1 link redundancy:

**Example:**
```

set interface LA/1 –haMonitor ON –throughput 29000

```
In case of a double-link failure between the primary appliance and the upstream switch the maximum Gi-LAN throughput that can be supported would fall to 20Gbps. A 29Gbps low threshold per the example above would result in a redundancy switchover event to the secondary appliance (which has not suffered similar link failures) so that Gi-LAN traffic is not affected.

## Route Monitors

In addition to LACP redundancy, route monitor checks might be configured and associated with the HA pair configuration. Route monitor checks can be useful to detect failures between the NetScaler appliance and the next-hop routers, especially if said routers are not directly connected but through an upstream switch.

A typical HA route monitor configuration per the sample Gi-LAN in section 2.5.1 is outlined below:
```

add route 192.168.1.0 255.255.255.0 192.168.1.1 -msr ENABLED -monitor arp
add route 192.168.2.0 255.255.255.0 192.168.2.1 -msr ENABLED -monitor arp
bind HA node -routeMonitor 192.168.1.0 255.255.255.0
bind HA node -routeMonitor 192.168.2.0 255.255.255.0
‘‘‘

TCP optimization configuration

September 21, 2020

Before configuring TCP optimization, apply the following basic configuration settings on the NetScaler appliance:

Initial configuration:

```
enable ns feature LB IPv6PT
enable ns mode FR L3 USIP MBF Edge USNIP PMTUD
disable ns feature SP
disable ns mode TCPB
set lb parameter -preferDirectRoute NO
set lb parameter -vServerSpecificMac ENABLED
set l4param -l2ConnMethod Vlan
set rsskeytype -rsstype SYMMETRIC
set ns param -useproxyport DISABLED
<!--NeedCopy-->
```

> **Note**
>
> Restart the NetScaler appliance if you change the `rsskeytype` system parameter.

TCP termination

For NetScaler T1 to apply TCP optimization it needs to first terminate incoming TCP traffic. Towards this end, a wildcard TCP vserver should be created and configured to intercept ingress traffic and then forward it to the Internet router.

Static or dynamic routing environment

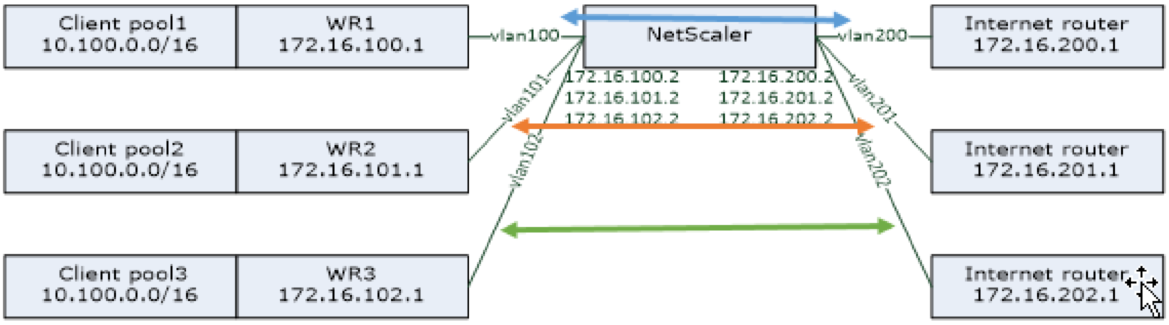
For environments with static or dynamic routing in place, vserver can rely on routing table info to forward packets towards internet router. Default route must point to the internet router and also routing entries for client subnets towards wireless router should be in place:

Example:

```
add lb vserver vsrv-wireless TCP * * -persistenceType NONE -
   Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER(\"vsrv-wireless
   \").STATE.EQ(UP)" -m IP -cltTimeout 9000
add route 0.0.0.0 0.0.0.0 192.168.2.1
add route 10.0.0.0 255.0.0.0 192.168.1.1
<!--NeedCopy-->
```

VLAN-to-VLAN (PBR) environment

There are customer environments where subscriber traffic is segmented to multiple flows and needs to be forwarded to different routers based on incoming traffic parameters. Policy Based Routing (PBR) can be used to route packets based on incoming packet parameters, such as VLAN, MAC address, Interface, source IP, source port, destination IP address, and destination port.


Client pool1
10.100.0.0/16
WR1
172.16.100.1
vlan100
NetScaler
vlan200
Internet router
172.16.200.1
vlan101
172.16.100.2
172.16.101.2
172.16.102.2
172.16.200.2
172.16.201.2
172.16.202.2
vlan201
Client pool2
10.100.0.0/16
WR2
172.16.101.1
vlan102
vlan202
Internet router
172.16.201.1
Client pool3
10.100.0.0/16
WR3
172.16.102.1
Internet router
172.16.202.1

Example:

```
add lb vserver vsrv-wireless TCP * * -m IP -l2Conn ON -listenpolicy "CLIENT.VLAN.ID.EQ(100) || CLIENT.VLAN.ID.EQ(101) || CLIENT.VLAN.ID.EQ(102)"

add ns pbr pbr-vlan100-to-vlan200 ALLOW -vlan 100 -nexthop 172.16.200.1

add ns pbr pbr-vlan101-to-vlan201 ALLOW -vlan 101 -nexthop 172.16.201.1

add ns pbr pbr-vlan102-to-vlan202 ALLOW -vlan 102 -nexthop 172.16.202.1
<!--NeedCopy-->
```

Using Policy Based Routing to route TCP optimized traffic is a new feature added in release 11.1 50.10. For previous releases, having multiple "mode MAC" vserver entities per VLAN is an alternative solution for multi-VLAN environments. Each vserver has a bound service representing the internet router for the particular flow.

Example:

```
add server internet_router_1 172.16.200.1

add server internet_router_2 172.16.201.1

add server internet_router_3 172.16.202.1

add service svc-internet-1 internet_router_1 TCP * -usip YES -useproxyport NO

add service svc-internet-2 internet_router_2 TCP * -usip YES -useproxyport NO

add service svc-internet-3 internet_router_3 TCP * -usip YES -useproxyport NO

```

```
bind service svc-internet-1 -monitorName arp

bind service svc-internet-2 -monitorName arp

bind service svc-internet-3 -monitorName arp

add lb vserver vsrv-wireless-1 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(100) && SYS.VSERVER(\"vsrv-wireless-1\").STATE.EQ(UP)" -m MAC -l2Conn ON

add lb vserver vsrv-wireless-2 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(101) && SYS.VSERVER(\"vsrv-wireless-2\").STATE.EQ(UP)" -m MAC -l2Conn ON

add lb vserver vsrv-wireless-3 TCP * * -Listenpolicy "CLIENT.VLAN.ID.EQ(102) && SYS.VSERVER(\"vsrv-wireless-3\").STATE.EQ(UP)" -m MAC -l2Conn ON

bind lb vserver vsrv-wireless-1 svc-internet-1

bind lb vserver vsrv-wireless-2 svc-internet-2

bind lb vserver vsrv-wireless-3 svc-internet-3
<!--NeedCopy-->
```

> **Note**
>
> The vserver mode is MAC in contrast to previous examples where it is mode IP. This is required to retain the destination IP information when we have service(s) bound to vserver. Also, the additional PBR configuration need to route non-optimized traffic.

TCP Optimization

Out-of-the-box NetScaler TCP termination is configured for TCP pass-through functionality. TCP pass-through essentially means that NetScaler T1 may transparently intercept a client-server TCP stream but does not retain separate client/server buffers or otherwise apply any optimization techniques.

To enable TCP optimization a TCP profile, named as nstcpprofile, is used to specify TCP configurations that is used if no TCP configurations are provided at the service or virtual server level and it should be modified as follows:

Command:

```
add ns tcpProfile nstcpprofile -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
```

```
bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT
<!--NeedCopy-->
```

> **Note:**
>
> If there is not any profile explicitly created and bound to vserver and service, the profile nstcp_default_profile is bound by default.
>
> In case of multiple TCP profiles requirement, extra TCP profiles can be created and associated with the appropriate virtual server

Command:

```
add ns tcpProfile custom_profile -WS ENABLED -SACK ENABLED -WSVal 8 -mss 1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -bufferSize 4000000 -flavor BIC -dynamicReceiveBuffering ENABLED -KA ENABLED -sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED -ecn ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -rstMaxAck enABLED -tcpmode ENDPOINT

set lb vserver vsrv-wireless -tcpProfileName custom_profile
<!--NeedCopy-->
```

> **Note:**
>
> For deployments with vserver -m MAC and service, same profile should be associated with service.

```
set service svc-internet -tcpProfileName custom_profile
<!--NeedCopy-->
```

TCP optimization capabilities

Most of the relevant TCP optimization capabilities of a NetScaler appliance are exposed through a corresponding TCP profile. Typical CLI parameters that should be considered when creating a TCP profile are the following:

1. **Window Scaling (WS)**: TCP Window scaling allows increasing the TCP receive window size beyond 65535 bytes. It helps improving TCP performance overall and specially in high bandwidth and long delay networks. It helps with reducing latency and improving response time over TCP.
2. **Selective acknowledgment (SACK)**: TCP SACK addresses the problem of multiple packet loss which reduces the overall throughput capacity. With selective acknowledgement the receiver

can inform the sender about all the segments which are received successfully, enabling sender to only retransmit the segments which were lost. This technique helps T1 improve overall throughput and reduce the connection latency.

3. **Window Scaling Factor (WSVal)**: Factor used to calculate the new window size. It must be configured with a high value in order to allow the advertised window by NS to be at least equal to the buffer size.
4. **Maximum Segment Size (MSS)**: MSS of a single TCP segment. This value depends on the MTU setting on intermediate routers and end clients. A value of 1460 corresponds to an MTU of 1500.
5. **maxBurst**: Maximum number of TCP segments allowed in a burst.
6. **Initial Congestion Window size(initialCwnd)**: TCP initial congestion window size determines the number of bytes which can be outstanding in beginning of the transaction. It enables T1 to send those many bytes without bothering for congestion on the wire.
7. **Maximum OOO packet queue size(oooQSize)**: TCP maintains Out Of Order queue to keep the OOO packets in the TCP communication. This setting impacts system memory if the queue size is long as the packets need to be kept in runtime memory. Thus this needs to be kept at optimized level based on the kind of network and application characteristics.
8. **Minimum RTO(minRTO)**: The TCP retransmission timeout is calculated on each received ACK based on internal implementation logic. The default retransmission timeout happens at 1 second to start with and this can be tweaked with this setting. For second retransmission of these packets RTO will be calculated by N*2 and then N*4 … N*8… goes on till last retransmission attempt.
9. **bufferSize / sendBuffsize**: these refer to the maximum amount of data that the T1 may receive from the server and buffer internally without sending to the client. They should be set to a value larger (at least double) than the Bandwidth Delay Product of the underlying transmission channel.
10. **flavor**: this refers to the TCP congestion control algorithm. Valid values are Default, BIC, CUBIC, Westwood and Nile.
11. **Dynamic receive buffering**: allows the receive buffer to be adjusted dynamically based on memory and network conditions. It will fill up the buffer as much as it's required to keep the client's download pipe full instead of filling up, by reading ahead from server, a fixed size buffer, as latter is specified in TCP profile and typically based on criteria such as 2*BDP, for a connection. NetScaler T1 monitors the network conditions to the client and estimates how much it should read ahead from the server.
12. **Keep-Alive (KA)**: Send periodic TCP keep-alive (KA) probes to check if peer is still up.
13. rstWindowAttenuate: Defending TCP against spoofing attacks. It will reply with corrective ACK when a sequence number is invalid.
14. **rstMaxAck**: Enable or disable acceptance of RST that is out of window yet echoes highest ACK sequence number.
15. **spoofSynDrop**: Drop of invalid SYN packets to protect against spoofing.

16. **Explicit Congestion Notification(ecn)**: It sends notification of the network congestion status to the sender of the data and takes corrective measures for data congestion or data corruption.
17. **Forward RTO-Recovery**: In case of spurious retransmissions, the congestion control configurations are reverted to their original state.
18. **TCP maximum congestion window (maxcwnd)**: TCP maximum congestion window size that is user configurable.
19. **Forward acknowledgment (FACK)**: To avoid TCP congestion by explicitly measuring the total number of data bytes outstanding in the network, and helping the sender (either T1 or a client) control the amount of data injected into the network during retransmission timeouts.
20. **tcpmode**: TCP optimization modes for specific profile.
21. **tcpmode**: TCP optimization modes for specific profile. There are two TCP optimization modes - Transparent and Endpoint.

- Endpoint. In this mode, the appliance manages the client and server connections seperately.
- Transparent. In the transparent mode the clients need to access the servers directly, with no intervening virtual server. The server IP addresses must be public because the clients need to be able to access them. In the example shown in the following figure, a NetScaler appliance is placed between the client and the server, so the traffic must pass through the appliance.

Analytics and Reporting

September 21, 2020

The TCP Speed Reporting is a NetScaler feature which extracts TCP connection statistics, as a measure of TCP download and upload performance, and is utilized in TCP Insight reports of the Citrix Application Delivery Management (ADM). To achieve this, NetScaler monitors each TCP connection, locates packet bursts on an idle time-out basis and reports key metrics (such as byte count, retransmitted byte count and duration) for the identified maximum burst. TCP Speed Reporting feature is enabled by default, support both TCP and HTTP vServers and depends on the Appflow/ULFD reporting infrastructure.

Real-time Statistics

September 21, 2020

The stat command might be used to verify that TCP optimization is properly applied:

Command:

```
> stat lb vserver vsrv-wireless
Virtual Server Summary
                      vsvrIP  port     Protocol        State    Health
   actSvcs
vsrv...eless               *     0          TCP           UP       100
           1

           inactSvcs
vsrv...eless       0
Virtual Server Statistics
                                          Rate (/s)
   Total
Vserver hits                                       0
   10
Requests                                           0
    0
Responses                                          0
    0
Request bytes                                      0
   1580
Response bytes                                     0
   532594360
Total Packets rcvd                                 0
   216463
Total Packets sent                                 0
   369898
Current client connections                        --
    0
Current Client Est connections                    --
    0
Current server connections                        --
    0
Requests in surge queue                           --
    0
Requests in vserver's surgeQ                      --
    0
Requests in service's surgeQs                     --
    0
Spill Over Threshold                              --
    0
Spill Over Hits                                   --
    0
Labeled Connection                                --
          0
Push Labeled Connection                           --
```

```
    0
Deferred Request                                   0
    0
Invalid Request/Response                          --
    0
Invalid Request/Response Dropped                  --
    0
Bound Service(s) Summary
                          IP  port         Type        State     Hits
    Hits/s
svc-internet     192.168.2.2     0          TCP           UP       10
         0/s

                 Req    Req/s      Rsp    Rsp/s Throughp ClntConn
   SurgeQ
svc-internet       0      0/s        0      0/s        0        0
           0
                  SvrConn   ReuseP  MaxConn ActvTran  SvrTTFB     Load
svc-internet         0        0        0        0        0        0
```

The Total counters should constantly increase for an operational system. In addition, the Rate counters should be non-zero.

> **Note**
>
> The preceding output is from an operational yet idle lab system, explaining the zero rate.

SNMP

September 21, 2020

SNMP agent can be queried for system specific information from a remote device (SNMP Manager). Based on the query, the agent searches for the equal object identifier (OID) in the management information base (MIB) for the data requested and sends the information to the SNMP Manager. The following are the most useful SNMP OIDs for Telco deployments:

Memory

- **resMemUsage (1.3.6.1.4.1.5951.4.1.1.41.2)**

Percentage of memory utilization on NetScaler.

Packet Engine CPU

- **resCpuUsage (1.3.6.1.4.1.5951.4.1.1.41.1)**

CPU utilization percentage.

- **nsCPUTable (1.3.6.1.4.1.5951.4.1.1.41.6)**

This table contains information about each CPU in NetScaler.

Indexed on: nsCPUname

- **nsCPUname (1.3.6.1.4.1.5951.4.1.1.41.6.1.1)**

The name of the CPU.

- **nsCPUusage (1.3.6.1.4.1.5951.4.1.1.41.6.1.2)**

CPU utilization percentage.

Throughput

- **allNicTotRxMbits (1.3.6.1.4.1.5951.4.1.1.71.1)**

Number of megabits received by the NetScaler appliance.

- **allNicTotTxMbits (1.3.6.1.4.1.5951.4.1.1.71.2)**

Number of megabits transmitted by the NetScaler appliance.

- **ipTotRxPkts (1.3.6.1.4.1.5951.4.1.1.43.25)**

IP packets received.

- **ipTotRxMbits (1.3.6.1.4.1.5951.4.1.1.43.27)**

Megabits of IP data received.

- **ipTotTxPkts (1.3.6.1.4.1.5951.4.1.1.43.28)**

IP packets transmitted.

- **ipTotTxMbits (1.3.6.1.4.1.5951.4.1.1.43.30)**

Megabits of IP data transmitted.

Connections

Active connections:

- **tcpActiveServerConn (1.3.6.1.4.1.5951.4.1.1.46.8)**

Connections to a server currently responding to requests.

Total connections:

- **tcpCurServerConn (1.3.6.1.4.1.5951.4.1.1.46.1)**

Server connections, including connections in the Opening, Established, and Closing state.

- **tcpCurClientConn (1.3.6.1.4.1.5951.4.1.1.46.2)**

Client connections, including connections in the Opening, Established, and Closing state.

Note: Because of SYN-Cookie, this doesn’t include Client in Opening state

- **tcpTotZombieCltConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.26)**

Client connections that are flushed because the client has been idle for some time.

- **tcpTotZombieSvrConnFlushed (1.3.6.1.4.1.5951.4.1.1.46.27)**

Server connections that are flushed because there have been no client requests in the queue for some time.

Errors

- **tcpErrSynGiveUp (1.3.6.1.4.1.5951.4.1.1.46.37)**

Attempts to establish a connection on the NetScaler that timed out.

- **tcpErrRetransmitGiveUp (1.3.6.1.4.1.5951.4.1.1.46.60)**

Number of times NetScaler terminates a connection after retransmitting the packet seven times on that connection.Retrasnmission happens when recieving end doesn’t acknowledges the packet.

- **ifInDiscards (1.3.6.1.2.1.2.2.1.13)**

The number of inbound packets which were chosen to be discarded even though no errors had been detected to prevent their being deliverable to a higher-layer protocol. One possible reason for discarding such a packet could be to free up buffer space.

- **ifOutDiscards (1.3.6.1.2.1.2.2.1.19)**

The number of outbound packets which were chosen to be discarded even though no errors had been detected to prevent their being transmitted. One possible reason for discarding such a packet could be to free up buffer space.

- **ifErrTxOverflow (1.3.6.1.4.1.5951.4.1.1.54.1.36)**

Number of packets that have passed through the overflow queues, during transmission on the specified interface, since the NetScaler appliance was started or the interface statistics were cleared. This gets incremented only on congested ports.

Optimized/Bypass connections

- **tcpOptimizationEnabled (1.3.6.1.4.1.5951.4.1.1.46.131)**

Total number of connections enabled with TCP optimization.

- **tcpOptimizationBypassed (1.3.6.1.4.1.5951.4.1.1.46.132)**

Total number of connections bypassed TCP Optimization.

Technical Recipes

September 21, 2020

The NetScaler T1 models provide advanced features and a powerful policy configuration language that allow for evaluation of complex decision in runtime.

While it is not possible to evaluate all capabilities that are potentially unlocked by the T1000 features and policy configuration guide, technical receipes consider implementation of various requirements brought in by Telco operators. Feel free to re-use the "recipes" as is or adapt to your environment.

Per-user Connection Limit

The NetScaler T1 model can be configured to limit the number of connections per unique subscriber IP. With the below configuration, N concurrent TCP connections per IP (CLIENT.IP.SRC) is allowed. For every attempt for connection beyond the configured threshold, T1 sends an RST. For maximum 2 concurrent connections per user:

Command:

```
add stream selector streamSel_usrlimit CLIENT.IP.SRC
add ns limitIdentifier limitId_usrlimit -threshold 2 -mode CONNECTION -
    selectorName streamSel_usrlimit
add responder policy respPol_usrlimit "SYS.CHECK_LIMIT(\"
    limitId_usrlimit\")" RESET
bind lb vserver vsrv-wireless -policyName respPol_usrlimit -priority 1
    -gotoPriorityExpression END
<!--NeedCopy-->
```

Smooth Insertion/Deletion of Vserver

Many operators concern about TCP connections disruption when the NetScaler T1 model is activated inline for TCP optimization or when it is disabled for maintenance purposes. To avoid breaking existing connections when vserver is introduced, the following configuration needs to be applied before configuring or activating vserver for TCP optimization:

Command:

```
add ns acl acl-ingress ALLOW  - vlan 100
add forwardingSession fwd-ingress  - aclname acl-ingress
apply ns acls
<!--NeedCopy-->
```

Forwarding sessions are effective on top of routing (either static or dynamic or PBR) and create session entries for traffic that is routed (L3 mode). Any existing connection is handled by forwarding session due to corresponding sessions, and upon vserver introduction it starts capturing only new TCP connections.

ACLs can be configured to capture only specific ports like vserver, in order to avoid creating sessions for unnecessary traffic, which is memory consuming. Another option is to remove specific configuration after vserver activation.

For maintenance purposes, vserver should be disabled and its state appears as OUT OF SERVICE. When this happens, the vserver terminates all connections immediately by default. To make vserver to still serve the existing connections and not accept new, the following configuration should be applied:

Command:

```
set lb vserver vsrv-wireless  - downStateFlush DISABLED
<!--NeedCopy-->
```

New connections go through the routing table, and corresponding session entries are created due to forwarding sessions.

Policy-Based TCP Profiling

Policy-based TCP Profile selection allows operators to configure TCP profile dynamically for clients coming from different traffic domains (i.e. 3G or 4G). Some of the QoS metrics are different for these traffic domains, and in order to achieve better performance, you need to change some of the TCP parameter dynamically. Consider a case where clients coming from 3G and 4G hit same vserver and use same TCP profile, which have negative impact on some client's performance. AppQoE functionality can classify these clients and dynamically change TCP profile on vserver.

Example:

```
enable feature AppQoE

add ns tcpProfile nstcpprofile1 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 4000000 -flavor BIC -KA ENABLED -
    sendBuffsize 4000000 -rstWindowAttenuate ENABLED -spoofSynDrop
```

```
ENABLED -frto ENABLED -maxcwnd 1000000 -fack ENABLED -tcpmode
    ENDPOINT

add ns tcpProfile nstcpprofile2 -WS ENABLED -SACK ENABLED -WSVal 8 -mss
     1460 -maxBurst 15 -initialCwnd 16 -oooQSize 15000 -minRTO 800 -
    slowStartIncr 1 -bufferSize 128000 -flavor BIC -KA ENABLED -
    sendBuffsize 6000000 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -frto ENABLED -maxcwnd 64000 -fack ENABLED -tcpmode ENDPOINT

add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "CLIENT.VLAN.ID.EQ(100)" -action
    action_1

add appqoe policy appqoe_3G -rule "CLIENT.VLAN.ID.EQ(200)" -action
    action_2

bind lb vserver vsrv-wireless -policyName appqoe_4G -priority 100

bind lb vserver vsrv-wireless -policyName appqoe_3G -priority 110
<!--NeedCopy-->
```

The NetScaler T1 model is capable to receive the subscriber information dynamically through Gx or Radius or Radius and Gx interface and apply different TCP profile on a per-subscriber basis.

Command:

```
add appqoe action action_1 -priority HIGH -tcpprofile nstcpprofile1

add appqoe action action_2 -priority HIGH -tcpprofile nstcpprofile2

add appqoe policy appqoe_4G -rule "SUBSCRIBER.RULE_ACTIVE(\"3G\")" -
    action action_1

add appqoe policy appqoe_3G -rule "SUBSCRIBER.RULE_ACTIVE(\"4G\")" -
    action action_2
<!--NeedCopy-->
```

For integration of the NetScaler T1 model with operator control-plane network, see Telco Subscriber Management.

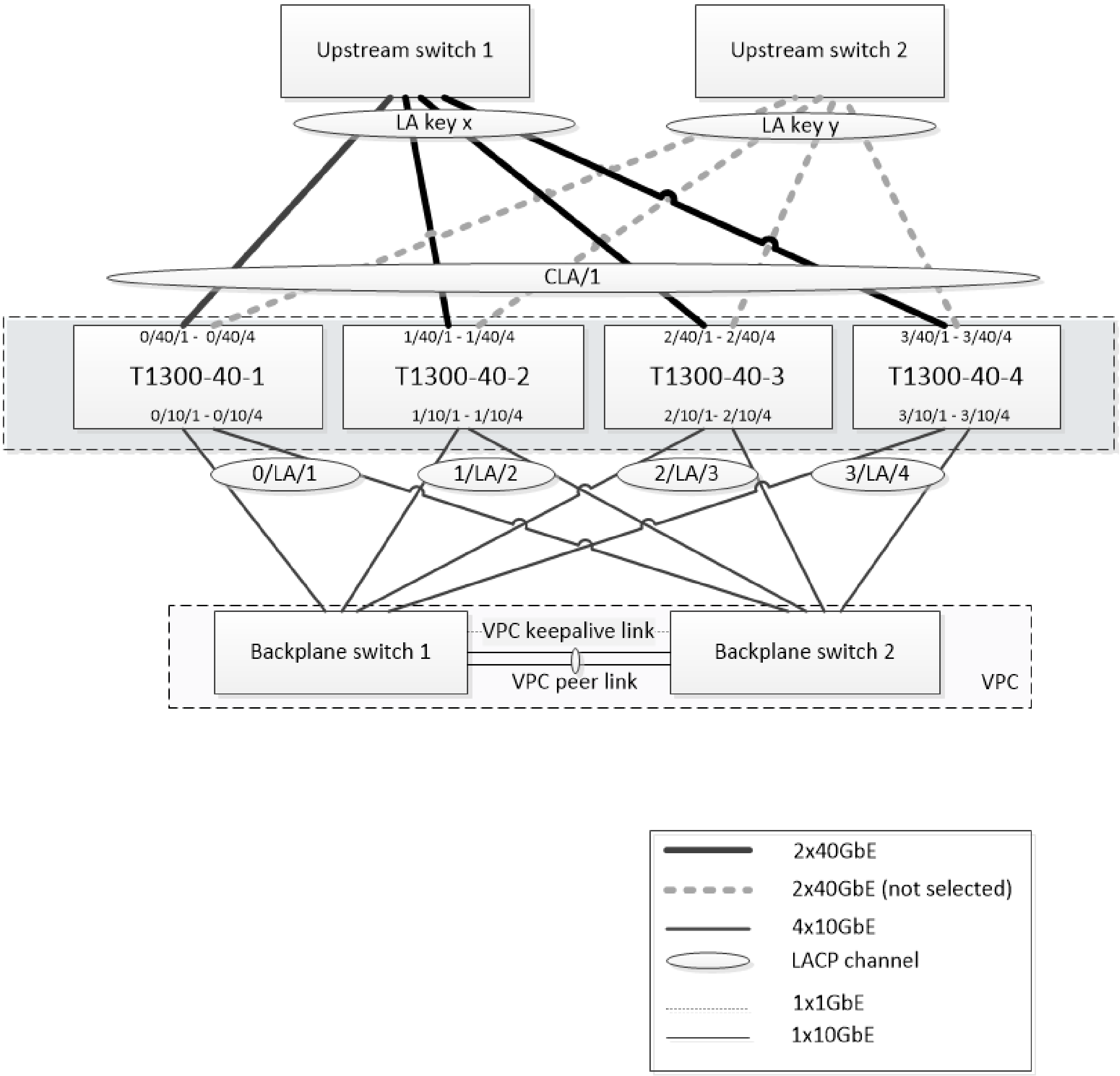
Scalability

September 21, 2020

Because TCP optimization is resource intensive, a single NetScaler appliance, even a high end –appliance, might not be able to sustain high Gi-LAN throughputs. To expand the capacity of your network, you can deploy NetScaler appliances in an N+1 cluster formation. In a cluster deployment, the NetScaler appliances work together as a single system image. The client traffic is distributed across the cluster nodes with the help of external switch device.

Topology

Figure 1 is an example of a cluster consisting of four T1300-40G nodes.


Upstream switch 1
Upstream switch 2
LA key x
LA key y
CLA/1
0/40/1 - 0/40/4
T1300-40-1
0/10/1 - 0/10/4
1/40/1 - 1/40/4
T1300-40-2
1/10/1 - 1/10/4
2/40/1 - 2/40/4
T1300-40-3
2/10/1- 2/10/4
3/40/1 - 3/40/4
T1300-40-4
3/10/1 - 3/10/4
0/LA/1
1/LA/2
2/LA/3
3/LA/4
Backplane switch 1
VPC keepalive link
VPC peer link
Backplane switch 2
VPC
2x40GbE
2x40GbE (not selected)
4x10GbE
LACP channel
1x1GbE
1x10GbE

The setup shown in Figure 1 has the following properties:

1. All cluster nodes belong to the same network (also known as an L2 cluster).
2. Data plane and backplane traffic are handled by different switches.
3. Assuming Gi-LAN throughput is 200 Gbps and that a T1300-40G appliance can sustain 80Gbps of throughput, we need three T1300-40G appliances. To provide redundancy in case of single cluster node failure, we deploy four appliances in total.
4. Each node will receive up to 67Gbps of traffic (50Gbps in normal operating conditions and 67Gbps in case of single cluster node failure), so it needs 2x40Gbps connections to the upstream switch. To provide redundancy in case of switch failure, we deploy a couple of upstream switches and double the number of connections.
5. Cluster Link Aggregation (CLAG) is used to distribute traffic across cluster nodes. A single CLAG handles both client and server traffic. Link Redundancy is enabled on the CLAG, so only one "subchannel" is selected at any given time and handles the traffic. If some link fails or throughput falls below specified threshold, the other subchannel is selected.
6. The upstream switch performs symmetric port-channel load balancing (for example, source-dest-ip-only algorithm of Cisco IOS 7.0(8) N1(1)) so that forward and reverse traffic flows are handled by the same cluster node. This property is desirable because it eliminates packet re-ordering, which would degrade TCP performance.
7. Fifty percent of data traffic is expected to be steered to backplane, which means each node will steer up to 34Gbps to other cluster nodes (25Gbps in normal operating conditions and 34Gbps in case of single cluster node failure). Thus, each node needs at least 4x10G connections to the backplane switch. To provide redundancy in case of switch failure, we deploy a couple of backplane switches and double the number of connections. Link redundancy is not currently supported for backplane, so Cisco VPC or equivalent technology is desired to achieve switch-level redundancy.
8. MTU size of steered packets is 1578 bytes, so backplane switches must support an MTU more than 1500 bytes.

Note: The design depicted in Figure 1 is also applicable to T1120 and T1310 appliances. For T1310 we would use 40GbE interfaces for the backplane connections, since it lacks 10GbE ports.

Note: While this document uses Cisco VPC as an example, if working with non-Cisco switches alternate equivalent solutions could be used, such as Juniper's MLAG.

Note: While other topologies such as ECMP instead of CLAG are possible, they are not currently supported for this particular use case.

Configuring TCP Optimization in a NetScaler T1000 Cluster

After physical installation, physical connectivity, software installation, and licensing are completed, you can proceed with the actual cluster configuration. The configurations described below apply to the cluster depicted in Figure 1.

Note: For more information about cluster configuration, see "Setting up a NetScaler cluster."

Assume that the four T1300 nodes in Figure 1 have the following NSIP addresses:

Four T1300 nodes with NSIP address:

```
T1300-40-1: 10.102.29.60
T1300-40-2: 10.102.29.70
T1300-40-3: 10.102.29.80
T1300-40-4: 10.102.29.90
```

The cluster will be managed through the cluster IP (CLIP) address, which is assumed to be 10.78.16.61.

Setting Up the Cluster

To begin configuring the cluster shown in Figure 1, log on to the first appliance that you want to add to the cluster (for example, T1300-40-1) and do the following.

1. At the command prompt, enter the following commands:

Command:

```
> add cluster instance 1
> add cluster node 0 10.102.29.60 -state ACTIVE
> add ns ip 10.102.29.61 255.255.255.255 -type clip
> enable cluster instance 1
> save ns config
> reboot  - warm
```

2. After the appliance restarts, connect to the Cluster IP (CLIP) address and add the rest of the nodes to the cluster:

Command:

```
> add cluster node 1 10.102.29.70 -state ACTIVE
> add cluster node 2 10.102.29.80 -state ACTIVE
> add cluster node 3 10.102.29.90  - state ACTIVE
> save ns config
```

3. Connect to the NSIP address of each of the newly added nodes and join the cluster:

Command:

```
> join cluster -clip 10.102.29.61 -password nsroot
> save ns config
> reboot  - warm
```

4. After the nodes restart, proceed with backplane configuration. On the cluster IP address, enter the following commands to create an LACP channel for the backplane link of each cluster node:

Command:

```
> set interface 0/10/[1-8]  - lacpkey 1  - lacpmode ACTIVE
> set interface 1/10/[1-8]  - lacpkey 2  - lacpmode ACTIVE
> set interface 2/10/[1-8]  - lacpkey 3  - lacpmode ACTIVE
> set interface 3/10/[1-8]  - lacpkey 4  - lacpmode ACTIVE
```

5. Similarly, configure dynamic LA and VPC on the backplane switches. Make sure the MTU of the backplane switch interfaces is at least 1578 bytes.

6. Verify the channels are operational:

Command:

```
> show channel 0/LA/1
> show channel 1/LA/2
> show channel 2/LA/3
> show channel 3/LA/4
```

7. Configure the cluster node backplane interfaces.

Command:

```
> set cluster node 0 -backplane 0/LA/1
> set cluster node 1 -backplane 1/LA/2
> set cluster node 2 -backplane 2/LA/3
> set cluster node 3  - backplane 3/LA/4
```

8. Check the cluster status and verify that the cluster is operational:

```
> show cluster instance
> show cluster node
```

For more information on cluster setup, see "Setting up aNetScaler cluster"

Distributing Traffic Across Cluster Nodes

After you have formed theNetScaler cluster, deploy Cluster Link Aggregation (CLAG) to distribute traffic across cluster nodes. A single CLAG link will handle both client and server traffic.

On the cluster IP address, execute the following commands to create the Cluster Link Aggregation (CLAG) group shown in Figure 1:

Command:

```
> set interface 0/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 3/40/[1-4] -lacpMode active -lacpKey 5 -lagType Cluster
```

Configure dynamic link aggregation on the external switches.

Then, enable Link Redundancy as follows:

Code:

```
> set channel CLA/1 -linkRedundancy ON -lrMinThroughput 240000
```

Finally, check the channel status by entering:

Command:

```
> show channel CLA/1
```

The channel should be UP and the actual throughput should be 320000.

For more information about cluster link aggregation, see the following topics:

- Dynamic Cluster Link Aggregation
- Link Redundancy in a Cluster with LACP.

Because we will be using MAC-based forwarding (MBF), configure a linkset and bind it to the CLAG group as follows:

Command:

```
> add linkset LS/1
> bind linkset LS/1 -ifnum CLA/1
```

More information about linksets, see the following topics:

- Configuring Linksets
- Using Cluster LA Channel with Linksets

Configuring VLAN and IP Addresses

We will be using striped IP configuration, which means that IP addresses are active on all nodes (default setting). See "Striped, Partially Striped, and Spotted Configurations" for more information about this topic.

1. Add the ingress and egress SNIPs:

Command:

```
> add ns ip 172.16.30.254 255.255.255.0  - type SNIP
> add ns ip 172.16.31.254 255.255.255.0  - type SNIP
> add ns ip6 fd00:172:16:30::254/112  - type SNIP
> add ns ip6 fd00:172:16:31::254/112  - type SNIP
```

2. Add the corresponding ingress and egress VLANs:

Command:

```
> add vlan 30 -aliasName wireless
> add vlan 31 -aliasName internet
```

3. Bind VLANs with IPs and linkset:

Command:

```
> bind vlan 31 -ifnum LS/1 -tagged
> bind vlan 30 -ifnum LS/1 -tagged
> bind vlan 30 -IPAddress 172.16.30.254 255.255.255.0
> bind vlan 31 -IPAddress 172.16.31.254 255.255.255.0
> bind vlan 30 -IPAddress fd00:172:16:30::254/112
> bind vlan 31 -IPAddress fd00:172:16:31::254/112
```

More ingress and egress VLANs can be added if needed.

Configuring TCP Optimization

At this point, we have applied all cluster specific commands. To complete the configuration, follow the steps described in TCP optimization configuration".

Configuring Dynamic Routing

A NetScaler cluster can be integrated into the dynamic routing environment of the customer's network. Following is an example of dynamic routing configuration using BGP routing protocol (OSPF is also supported).

1. From the CLIP address, enable BGP and dynamic routing on ingress and egress IP addresses:

Command:

```
> enable ns feature bgp
> set ns ip 172.16.30.254  - dynamicRouting ENABLED
> set ns ip 172.16.31.254  - dynamicRouting ENABLED
```

2. Open vtysh and configure BGP for the egress side:

Code:

```
> shell
root@ns# vtysh
ns# configure terminal
ns(config)# router bgp 65531
ns(config-router)# network 10.0.0.0/24
ns(config-router)# neighbor 172.16.31.100 remote-as 65530
ns(config-router)# neighbor 172.16.31.100 update-source 172.16.31.254
ns(config-router)# exit
ns(config)# ns route-install propagate
ns(config)# ns route-install default
ns(config)# ns route-install bgp
ns(config)# exit
```

3. Configure the egress-side BGP peer to advertise the default route to the NetScaler cluster. For example:

Command:

```
router bgp 65530
 bgp router-id 172.16.31.100
 network 0.0.0.0/0
 neighbor 172.16.31.254 remote-as 65531
```

4. Follow similar steps to configure the ingress side.

5. From vtysh verify that configuration is propagated to all cluster nodes, by entering:

Command:

```
ns# show running-config
```

6. Finally, log on to NSIP address of each cluster node and verify routes advertised from BGP peer:

Command:

```
> show route | grep BGP
```

Optimizing TCP Performance using TCP Nile

September 21, 2020

TCP uses the following optimization techniques and congestion control strategies (or algorithms) to avoid network congestion in data transmission.

Congestion Control Strategies

The Transmission Control Protocol (TCP) has long been used to establish and manage Internet connections, handle transmission errors, and smoothly connect web applications with client devices. But network traffic has become more difficult to control, because packet loss does not depend only on the congestion in the network, and congestion does not necessarily cause packet loss. Therefore, to measure congestion, a TCP algorithm should focus on both packet loss and bandwidth.

NILE Algorithm

Citrix Systems has developed a new congestion-control algorithm, NILE, a TCP optimization algorithm designed for high-speed networks such as LTE, LTE advanced and 3G. Nile addresses unique challenges caused by fading, random or congestive losses, link layer retransmissions and carrier aggregation.

The NILE algorithm:

- Bases queue-latency estimates on round-trip time measurements.
- Uses a congestion-window-increase function that is inversely proportional to the measured queue latency. This method results in approaching the network congestion point more slowly than does the standard TCP method, and reduces the packet losses during congestion.
- Can distinguish between random loss and congestion based loss on the network by using the estimated queue latency.

The telecom service providers can use the NILE algorithm in their TCP infrastructure to:

- Optimize mobile and long-distance networks— The NILE algorithm achieves higher throughput compared to standard TCP. This feature is especially important for mobile and long-distance networks.
- Decrease application perceived latency and enhance subscriber experience— The Nile algorithm uses packet loss information to determine whether the transmission-window size should be increased or decreased, and uses queuing delay information to determine the size of the increment or decrement. This dynamic setting of transmission-window size decreases the application latency on the network.

To configure NILE support using the command line interface

At the command prompt, type the following:

```
set ns tcpProfile <name> [-flavor NILE]
```

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

Configuring NILE support using the configuration utility

1. Navigate to **System**> **Profiles**> **TCP Profiles** and click **TCP profiles**.
2. From the **TCP Flavor** drop-down list, select **NILE**.

Example:

```
set ns tcpProfile tcpprofile1 -flavor NILE
<!--NeedCopy-->
```

Proportional Rate Recovery (PRR) Algorithm

TCP Fast Recovery mechanisms reduce web latency caused by packet losses. The new Proportional Rate Recovery (PRR) algorithm is a fast recovery algorithm that evaluates TCP data during a loss recovery. It is patterned after Rate-Halving, by using the fraction that is appropriate for the target window chosen by the congestion control algorithm. It minimizes window adjustment, and the actual window size at the end of recovery is close to the Slow-Start threshold (ssthresh).

TCP Fast Open (TFO)

TCP Fast Open (TFO) is a TCP mechanism that enables speedy and safe data exchange between a client and a server during TCP's initial handshake. This feature is available as a TCP option in the TCP profile bound to a virtual server of a NetScaler appliance. TFO uses a TCP Fast Open Cookie (a security cookie) that the NetScaler appliance generates to validate and authenticate the client initiating a TFO connection to the virtual server. By using the TFO mechanism, you can reduce an application's network latency by the time required for one full round trip, which significantly reduces the delay experienced in short TCP transfers.

How TFO works

When a client tries to establish a TFO connection, it includes a TCP Fast Open Cookie with the initial SYN segment to authenticate itself. If authentication is successful, the virtual server on the NetScaler appliance can include data in the SYN-ACK segment even though it has not received the final ACK segment of the three-way handshake. This saves up to one full round-trip compared to a normal TCP connection, which requires a three-way handshake before any data can be exchanged.

A client and a backend server perform the following steps to establish a TFO connection and exchange data securely during the initial TCP handshake.

1. If the client does not have a TCP Fast Open Cookie to authenticate itself, it sends a Fast Open Cookie request in the SYN packet to the virtual server on the NetScaler appliance.
2. If the TFO option is enabled in the TCP profile bound to the virtual server, the appliance generates a cookie (by encrypting the client's IP address under a secret key) and responds to the client with a SYN-ACK that includes the generated Fast Open Cookie in a TCP option field.
3. The client caches the cookie for future TFO connections to the same virtual server on the appliance.
4. When the client tries to establish a TFO connection to the same virtual server, it sends SYN that includes the cached Fast Open Cookie (as a TCP option) along with HTTP data.
5. The NetScaler appliance validates the cookie, and if the authentication is successful, the server accepts the data in the SYN packet and acknowledges the event with a SYN-ACK, TFO Cookie, and HTTP Response.

Note: If the client authentication fails, the server drops the data and acknowledges the event only with a SYN indicating a session timeout.

1. On the server side, if the TFO option is enabled in a TCP profile bound to a service, the NetScaler appliance determines whether the TCP Fast Open Cookie is present in the service to which it is trying to connect.
2. If the TCP Fast Open Cookie is not present, the appliance sends a cookie request in the SYN packet.
3. When the backend server sends the Cookie, the appliance stores the cookie in the server information cache.
4. If the appliance already has a cookie for the given destination IP pair, it replaces the old cookie with the new one.
5. If the cookie is available in the server information cache when the virtual server tries to reconnect to the same backend server by using the same SNIP address, the appliance combines the data in SYN packet with the cookie and sends it to the backend server.
6. The backend server acknowledges the event with both data and a SYN.

Note: If the server acknowledges the event with only a SYN segment, the NetScaler appliance immediately resends the data packet after removing the SYN segment and the TCP options from the original packet.

Configuring TCP Fast Open

To use the TCP Fast Open (TFO) feature, enable the TCP Fast Open option in the relevant TCP profile and set the TFO Cookie Timeout parameter to a value that suits the security requirement for that profile.

To enable or disable TFO by using the command line

At the command prompt, type one of the following commands to enable or disable TFO in a new or existing profile.

Note: The default value is DISABLED.

```
add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
unset tcpprofile <TCP Profile Name> - tcpFastOpen
<!--NeedCopy-->
```

Examples:

add tcpprofile Profile1 – tcpFastOpen
Set tcpprofile Profile1 – tcpFastOpen Enabled
unset tcpprofile Profile1 – tcpFastOpen

To set TCP Fast Open cookie timeout value by using the command line interface

At the command prompt, type:

```
set tcpparam –tcpfastOpenCookieTimeout <Timeout Value>
<!--NeedCopy-->
```

Example:

```
set tcpprofile –tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```

To configure the TCP Fast Open by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **TCP Fast Open** checkbox.
3. Click **OK** and then **Done**.

To Configure the TCP Fast Cookie timeout value by using the GUI

Navigate to **Configuration** > **System** > **Settings** > **Change TCP Parameters** and then **Configure TCP Parameters** page to set the TCP Fast Open Cookie timeout value.

TCP Hystart

A new TCP profile parameter, hystart, enables the Hystart algorithm, which is a slow-start algorithm that dynamically determines a safe point at which to terminate (ssthresh). It enables a transition to congestion avoidance without heavy packet losses. This new parameter is disabled by default.

If congestion is detected, Hystart enters a congestion avoidance phase. Enabling it gives you better throughput in high-speed networks with high packet loss. This algorithm helps maintain close to maximum bandwidth while processing transactions. It can therefore improve throughput.

Configuring TCP Hystart

To use the Hystart feature, enable the Cubic Hystart option in the relevant TCP profile.

To configure Hystart by using the command line interface (CLI)

At the command prompt, type one of the following commands to enable or disable Hystart in a new or existing TCP profile.

```
add tcpprofile <profileName> -hystart ENABLED
set tcpprofile <profileName> -hystart ENABLED
unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

Examples:

```
add tcpprofile Profile1 - tcpFastOpen
Set tcpprofile Profile1 - tcpFastOpen Enabled
unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```

To configure Hystart support by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **Cubic Hystart** check box.
3. Click **OK** and then **Done**.

Optimization Techniques

TCP uses the following optimization techniques and methods for optimized flow controls.

Policy based TCP Profile Selection

Network traffic today is more diverse and bandwidth-intensive than ever before. With the increased traffic, the effect that Quality of Service (QoS) has on TCP performance is significant. To enhance QoS, you can now configure AppQoE policies with different TCP profiles for different classes of network traffic. The AppQoE policy classifies a virtual server's traffic to associate a TCP profile optimized for a particular type of traffic, such as 3G, 4G, LAN, or WAN.

To use this feature, create a policy action for each TCP profile, associate an action with AppQoE policies, and bind the policies to the load balancing virtual servers.

Configuring Policy Based TCP Profile Selection

Configuring policy based TCP profile selection consists of the following tasks:

- Enabling AppQoE. Before configuring the TCP profile feature, you must enable the AppQoE feature.
- Adding AppQoE Action. After enabling the AppQoE feature, configure an AppQoE action with a TCP profile.
- Configuring AppQoE based TCP Profile Selection. To implement TCP profile selection for different classes of traffic, you must configure AppQoE policies with which your NetScaler appliance can distinguish the connections and bind the correct AppQoE action to each policy.
- Binding AppQoE Policy to Virtual Server. Once you have configured the AppQoE policies, you must bind them to one or more load balancing, content switching, or cache redirection virtual servers.

Configuring using the command line interface

To enable AppQoE by using the command line interface:

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

```
enable ns feature appqoe

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

To bind a TCP profile while creating an AppQoE action using the command line interface

At the command prompt, type the following AppQoE action command with tcpprofiletobind option.

Binding a TCP Profile:

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS |
    NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <
    string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth
     <positive_integer>] [-priqDepth <positive_integer>] [-
    dosTrigExpression <expression>] [-dosAction ( SimpleResponse |
    HICResponse )] [-tcpprofiletobind <string>]

show appqoe action
```

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

To configure an AppQoE policy by using the command line interface

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
<!--NeedCopy-->
```

To bind an AppQoE policy to load balancing, cache redirection or content switching virtual servers by using the command line interface

At the command prompt, type:

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <
    priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <
    priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <
    priority>
<!--NeedCopy-->
```

Example:

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle
    ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -
    slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -
    sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
    ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack ENABLED
    -tcpmode ENDPOINT

add appqoe action appact1 -priority HIGH -tcpprofile tcp1

add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
    action appact1

bind lb vserver lb2 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST

bind cs vserver cs1 -policyName apppol1 -priority 1 -
    gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

Configuring Policy based TCP Profiling using the GUI

To enable AppQoE by using the GUI

1. Navigate to **System**> **Settings**.
2. In the details pane, click **Configure Advanced Features.**
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

To configure AppQoE policy by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
3. To create a new action, click **Add**.
4. To modify an existing action, select the action, and then click **Edit**.
5. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):
    a) Name—name
    b) Action type—respondWith
    c) Priority—priority
    d) Policy Queue Depth—polqDepth
    e) Queue Depth—priqDepth
    f) DOS Action—dosAction

6. Click **Create**.

To bind AppQoE policy by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select a server and then click **Edit**.
2. In the **Policies** section and click (+) to bind an AppQoE policy.
3. In the **Policies** slider, do the following:
    a) Select a policy type as AppQoE from the drop-down list.
    b) Select a traffic type from the drop-down list.
4. In the **Policy Binding** section, do the following:
    a) Click **New** to create a new AppQoE policy.
    b) Click **Existing Policy** to select an AppQoE policy from the drop-down list.
5. Set the binding priority and click **Bind** to the policy to the virtual server.
6. Click **Done**.

SACK Block Generation

TCP performance slows down when multiple packets are lost in one window of data. In such a scenario, a Selective Acknowledgement (SACK) mechanism combined with a selective repeat retransmission policy overcomes this limitation. For every incoming out-of-order packet, you must generate a SACK block.

If the out-of-order packet fits in the reassembly queue block, insert packet info in the block, and set the complete block info as SACK-0. If an out-of-order packet does not fit into reassembly block, send the packet as SACK-0 and repeat the earlier SACK blocks. If an out-of-order packet is a duplicate and packet info is set as SACK-0 then D-SACK the block.

Note: A packet is considered as D-SACK if it is an acknowledged packet, or an out of order packet which is already received.

Client Reneging

A NetScaler appliance can handle client reneging during SACK based recovery.

Memory checks for marking end_point on PCB is not considering total available memory

In a NetScaler appliance, if the memory usage threshold is set to 75 percent instead of using the total available memory, it causes new TCP connections to bypass TCP optimization.

Unnecessary retransmissions due to missing SACK blocks

In a non-endpoint mode, when you send DUPACKS, if SACK blocks are missing for few out of order packets, triggers additional retransmissions from the server.

SNMP for number of connections bypassed optimization because of overload

The following SNMP ids have been added to a NetScaler appliance to track number of connections bypassed TCP optimization due to overload.

1. 1.3.6.1.4.1.5951.4.1.1.46.13 (tcpOptimizationEnabled). To track the total number of connections enabled with TCP optimization.
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). To track the total number of connections bypassed TCP Optimization.

Dynamic Receive Buffer

To maximize TCP performance, a NetScaler appliance can now dynamically adjust the TCP receive buffer size.

Troubleshooting Guidelines

September 21, 2020

Technical Support

All troubleshooting and escalation queries require a recent NetScaler techsupport bundle, which captures current configuration, firmware version installed, log files, outstanding cores, and others.

Example:

```
show techsupport

showtechsupport data collector tool - $Revision: #5 $!
...
All the data will be collected under
        /var/tmp/support/collector_P_192.168.121.117_18Jun2015_09_53
...
Archiving all the data into "/var/tmp/support/collector_P_192.168.121.117_18Jun2015_09_53.tar.gz" ....
Created a symbolic link for the archive with /var/tmp/support/support.tgz
/var/tmp/support/support.tgz  ---- points to ---> /var/tmp/support/collector_P_192.168.121.117_18Jun2015_09_53.tar.gz

After a techsupport bundle has been generated, it might be copied using SCP.
<!--NeedCopy-->
```

Traces

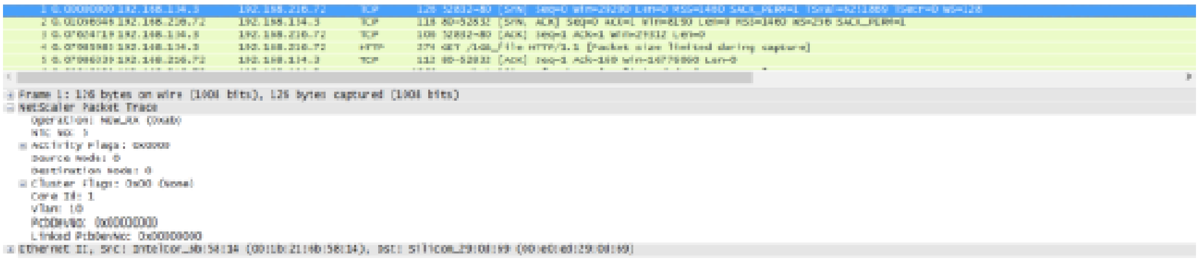
NetScaler TCP optimization issues normally require NetScaler traces to troubleshoot properly. Note that one should try to capture traces under similar conditions, i.e. on the same cell, during the same time of day, using the same user equipment and application, and others.

The start nstrace and stop nstrace commands might be used to capture traces:

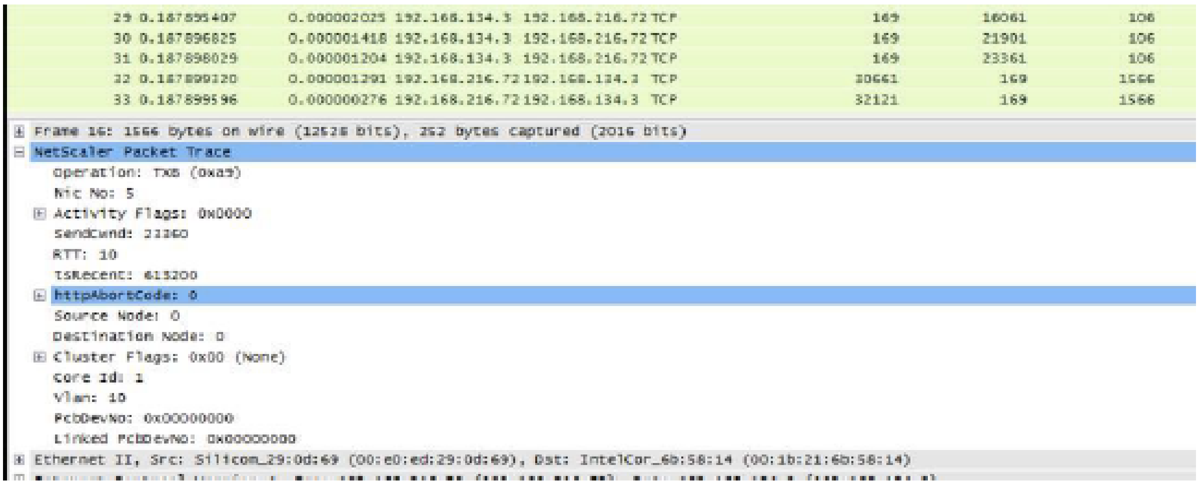
- It's strongly recommended that the appropriate filter is used to avoid capturing extraneous, unnecessary packets on the trace. For instance use start nstrace -filter 'IP == 10.20.30.40' to only capture packets being sent to or received from IP address 10.20.30.40, which is the user equipment IP address.
- Do not use the -tcpdump option, since it strips the nstrace headers which are required for debugging.

Trace Analysis

After a NetScaler trace has been captured, it might be viewed with Wireshark 1.12 or later. Verify that the captured traces include the appropriate NetScaler Packet Trace headers, as shown in the screen capture below:



The additional debug headers are also visible per the illustration below:



```
29 0.187895407   0.000002025 192.168.134.3  192.168.216.72 TCP   169   16061   106
30 0.187896825   0.000001418 192.168.134.3  192.168.216.72 TCP   169   21901   106
31 0.187898029   0.000001204 192.168.134.3  192.168.216.72 TCP   169   23361   106
32 0.187899120   0.000001291 192.168.216.72 192.168.134.3  TCP 30661     169  1566
33 0.187899596   0.000000276 192.168.216.72 192.168.134.3  TCP 32121     169  1566

⊞ Frame 16: 1566 bytes on wire (12528 bits), 252 bytes captured (2016 bits)
⊟ NetScaler Packet Trace
    Operation: TXB (0xa9)
    Nic No: 5
  ⊞ Activity Flags: 0x0000
    SendCwnd: 23360
    RTT: 10
    TsRecent: 613200
  ⊞ httpAbortCode: 0
    Source Node: 0
    Destination Node: 0
  ⊞ Cluster Flags: 0x00 (None)
    Core Id: 1
    Vlan: 10
    PcbDevNo: 0x00000000
    Linked PcbDevNo: 0x00000000
⊞ Ethernet II, Src: Silicom_29:0d:69 (00:e0:ed:29:0d:69), Dst: IntelCor_6b:58:14 (00:1b:21:6b:58:14)
```

Connection Table

When the issue is related to TCP optimization and it can be reproduced or it's on-going, it is best to get also the connection table when the issue occurs from the primary T1 node.

To get the table you shall need to switch to the BSD shell and run the following command:

```
shell
...

nscli -U 127.0.0.1:nsroot:nsroot show connectiontable -detail full link
     > /var/tmp/contable.log
<!--NeedCopy-->
```

> **Note**
>
> The command might be executed for a longer time and management CPU might be stressed at that time (depends on the number of connection table entries), but it's not service affecting.

Frequently Asked Questions

September 21, 2020

Timeouts

> **Important**
>
> Before using *any nsapimgr* knob, consult with Citrix Customer Support.

The following is a list of different idle connection timeouts that can be set on NetScaler T1 virtual servers and services. Idle timeout set for client or server connections at the vserver or service level are applicable only for the connections in TCP ESTABLISHED state and are idle.

- Load Balancing virtual server cltTimeout parameter specifies the time in seconds that a connection from a client to a Load Balancing virtual server must be idle, before the appliance closes the connection.
- Service svrTimeout parameter specifies the time in seconds that a connection from the appliance to a service or server must be idle, before the appliance closes the connection.
- Service cltTimeout parameter specifies the time in seconds that a connection from a client to a service must be idle, before the appliance closes the connection.

When a service is bound to a Load Balancing virtual server, then the cltTimeout for the Load Balancing virtual server takes precedence, and the service cltTimeout for service is ignored.

In case of there is not service bound to Load Balancing virtual server, global idle timeout, namely tcpServer, is used for server side connections. It can be configured as follows:

Command:

```
set ns timeout - tcpServer 9000
```

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

Connections in other state have different timeout values:

- Half open connections idle timeout: 120 seconds (hardcoded value)
- TIME_WAIT connections idle timeout: 40 seconds (hardcoded value)
- Half close connections idle timeout. By default it is 10s and can be configured between 1s and 600s using the snippet

Command:

```
 set ns timeout - halfclose 10
<!--NeedCopy-->
```

When half-close timeout is triggered, connection is moved to zombie state. When zombie timeout expires, zombie cleanup kicks in and T1 sends RST on both client and server side for given connection by default.

- Zombie timeout: Interval at which the zombie cleanup process must run to clean up inactive TCP connections. Default timeout value is 120s and can be configured between 1s and 600s.

Command:

```
set ns timeout - zombie 120
<!--NeedCopy-->
```

Maximum Segment Size Table

A NetScaler T1 appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted. Specific function is enabled by default without option to disable.

However, there is caveat as standard SYN cookies limit connections to the use of only eight Maximum Segment Size (MSS) values. If connection MMS does not match with any predefined value, it will pick up the next available lower value towards both client and server side.

The predefined TCP Maximum Segment Size (MSS) values are the following and can be configured through a new nsapimgr knob.

1460	1440	1330	1220	956	536	384	128

The new MSS table:

- Need not contain Jumbo-Frame support. Even though by default 8 values are reserved in the MSS table for jumbo frames, the table settings can be modified to include standard Ethernet-sized frames only.
- Should have 16 values
- Should have values in descending order
- Should include 128 as the last value

If the new MSS table is valid, the table is stored and the old values are switched out at the SYN-cookie rotation time. Otherwise the new table returns an error. Changes are applied to new connections while existing connections preserve the old MSS table until the connections expire or are terminated.

To display the current MSS table in a NetScaler appliance, type the following command.

Command:

```
>shell

#nsapimgr -d mss_table
```

EXample:

```
#nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1440,1330,1212,956,536,384,128
     }


Done.
```

To change the mss table, type the following command:

Command:

```
>shell

#nsapimgr -s mss_table=<16 comma seperated values>
```

Example:

```
#nsapimgr -ys mss_table=9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128

# nsapimgr -d mss_table

MSS table

{
 9176,9156,8192,7168,6144,4196,3072,2048,1460,1400,1330,1212,956,536,384,128 }


Done.
```

An example using standard Ethernet-sized values is depicted below:

Example:

```
#nsapimgr -ys mss_table=1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128

# nsapimgr -d mss_table

MSS table

{
 1460,1440,1420,1400,1380,1360,1340,1320,1300,1280,1260,1212,956,536,384,128 }


Done.
```

To make this change permanent even after the NetScaler appliance restarts, include the command "#nsapimgr -ys mss_table=<16 comma seperated values>" in the "/nsconfig/rc.netscaler" file. If the "rc.netscaler" file doesn't exist, create it under the "/nsconfig" folder, and then append the command.

Memory Overload Protection

A NetScaler Packet Processing Engine (PPE) starts bypassing connections from TCP optimization if the memory in use by that one PPE is more than a specified high watermark value. If a PPE memory utilization goes above ~2.6GB, then it starts bypassing *any new* connections from optimization. The existing connections (ones admitted for optimization previously) continues getting optimization. This watermark value has been purposefully selected and is not recommended for tuning.

> **Note**
>
> If you believe that there is a good reason to change that watermark value, contact Customer Support.

Support for Happy Eyeballs Clients

If the NetScaler appliance receives a SYN for a destination for which the state is unknown, the appliance first checks the reachability of the server and then acknowledges the client. This probing mechanism enables clients with dual IP stacks to discover the reachability of dual-stack internet servers. If the client discovers that both IPv6 and IPv4 access are available, it establishes a connection to the server that responds more quickly, and resets the other. For the connection for the NetScaler appliance receives a reset, it will reset the corresponding server side connection.

Note: This feature has no user configurable TCP settings to be disabled/enabled on the NetScaler appliance.

For more information on Happy Eyeballs support, see RFC 6555.

NetScaler Solutions

September 21, 2020

NetScaler solutions simplify the task of setting up frequently deployed configurations.

This section includes the following solutions

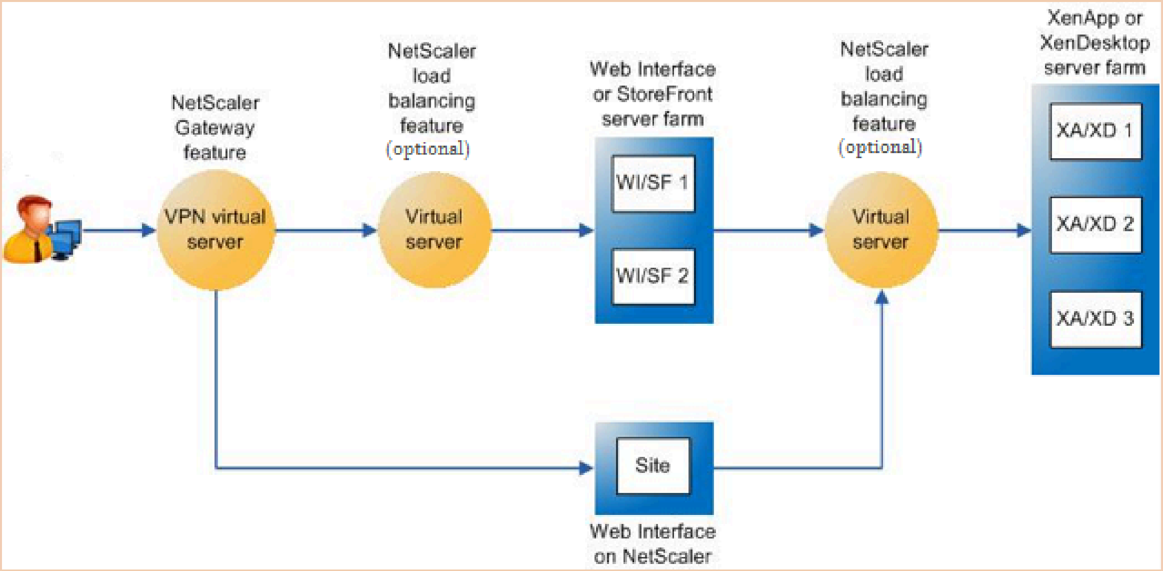
- Setting Up NetScaler for XenApp/XenDesktop
- Global Server Load Balancing (GSLB) Powered Zone Preference
- NetScaler in a Private Cloud Managed by Microsoft Windows Azure Pack and Cisco ACI

Setting up NetScaler for Citrix Virtual Apps and Desktops

September 21, 2020

A NetScaler appliance can provide load balanced, secure remote access to your Citrix Virtual Apps and Desktops applications. You can use the NetScaler load balancing feature to distribute traffic across the Citrix Virtual Apps and Desktops server, and the Citrix Gateway feature to provide secure remote access to the servers. NetScaler can also accelerate and optimize the traffic flow and offer visibility features that are useful for Citrix Virtual Apps and Desktops deployments.

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


NetScaler Gateway feature
VPN virtual server
NetScaler load balancing feature (optional)
Virtual server
Web Interface or StoreFront server farm
WI/SF 1
WI/SF 2
NetScaler load balancing feature (optional)
Virtual server
XenApp or XenDesktop server farm
XA/XD 1
XA/XD 2
XA/XD 3
Site
Web Interface on NetScaler

The above figure shows the components involved in this deployment:

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

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

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

Prerequisites

- Citrix Virtual Apps and Desktop servers are configured and available.
- Web Interface, StoreFront, or Web Interface on NetScaler servers are configured and available.
- You have a working knowledge of Citrix Gateway, NetScaler, Citrix Virtual Apps and Desktops, and StoreFront/Web Interface/Web Interface on NetScaler.
- Make sure that you have configured a virtual server and a service and bound the service to the virtual server.

Procedure:

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

Global Server Load Balancing (GSLB) Powered Zone Preference

September 21, 2020

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

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

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

For more information about configuring this solution, click here.

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

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

September 21, 2020

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

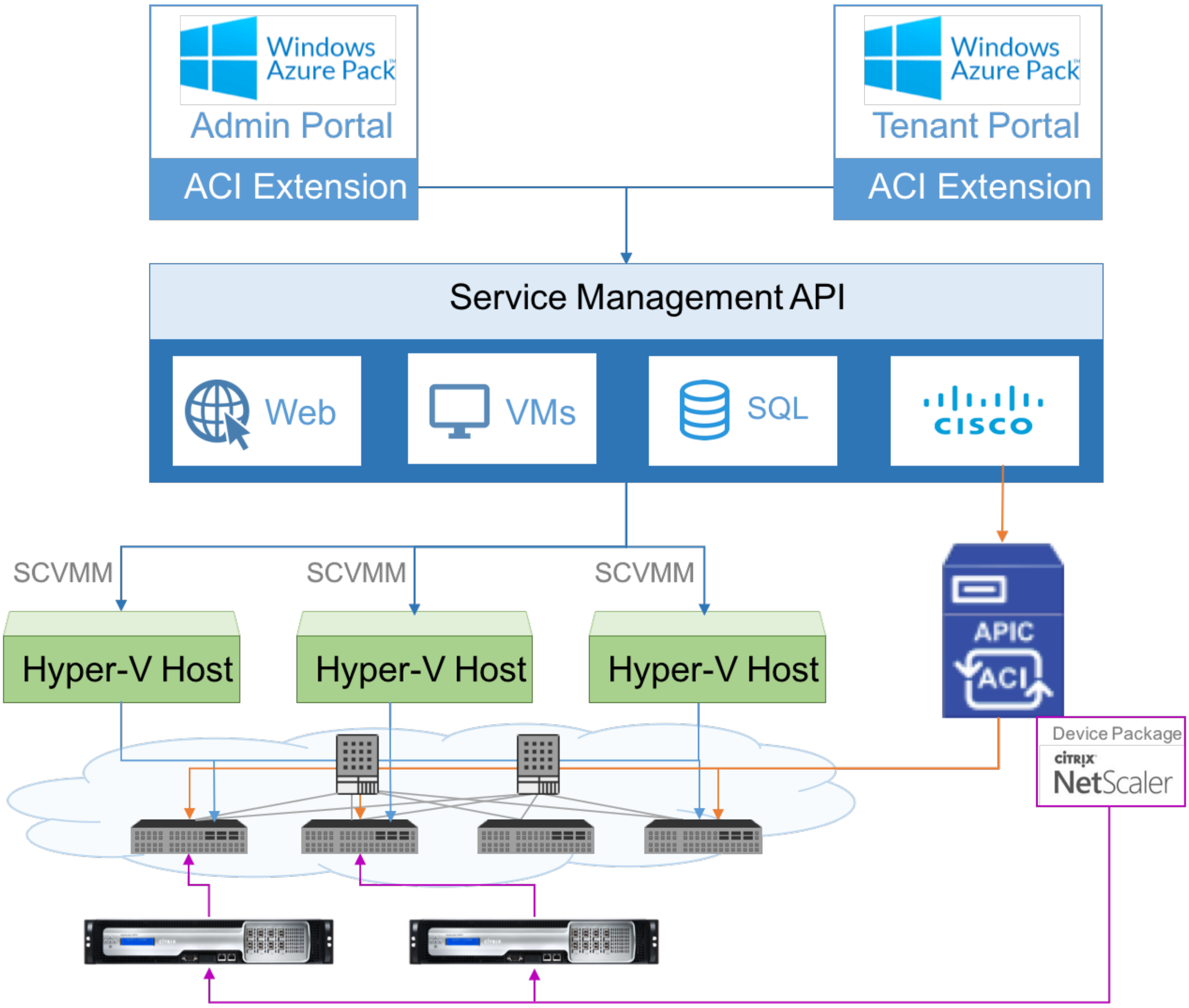
This solution involves many integration points, such as Windows Azure Pack (WAP) to Cisco APIC, Cisco APIC to System Center Virtual Machine Manager (SCVMM), and Cisco APIC to NetScaler. As a tenant in the private cloud, you can enable NAT, provision network services, and add a load balancer.

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

> **Important**
>
> - In this solution, the NetScaler appliance provides only basic load balancing.
> - Tenants can deploy multiple VIP addresses with different ports for the same network, but must ensure that the IP and port combination is unique.

> - The NetScaler device package supports only single-context deployment. Each Tenant gets a dedicated NetScaler instance.
> - WAP supports NetScaler MPX appliances and NetScaler VPX virtual appliances, including NetScaler VPX instances deployed on the NetScaler SDX platform.

The following illustration provides an overview of the solution:


Windows Azure Pack
Admin Portal
ACI Extension
Windows Azure Pack
Tenant Portal
ACI Extension
Service Management API
Web
VMs
SQL
cisco
SCVMM
SCVMM
SCVMM
Hyper-V Host
Hyper-V Host
Hyper-V Host
APIC
ACI
Device Package
citrix
NetScaler

Prerequisites

Make sure that:

- You have conceptual knowledge of Cisco ACI components and NetScalers.
  - For more information about Cisco ACI and its components, see the product documentation at: http://www.cisco.com/c/en/us/support/cloud-systems-management/application-policy-infrastructure-controller-apic/tsd-products-support-series-home.html.
  - For more information about the NetScalers, see the NetScaler product documentation at

  http://docs.citrix.com/.

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

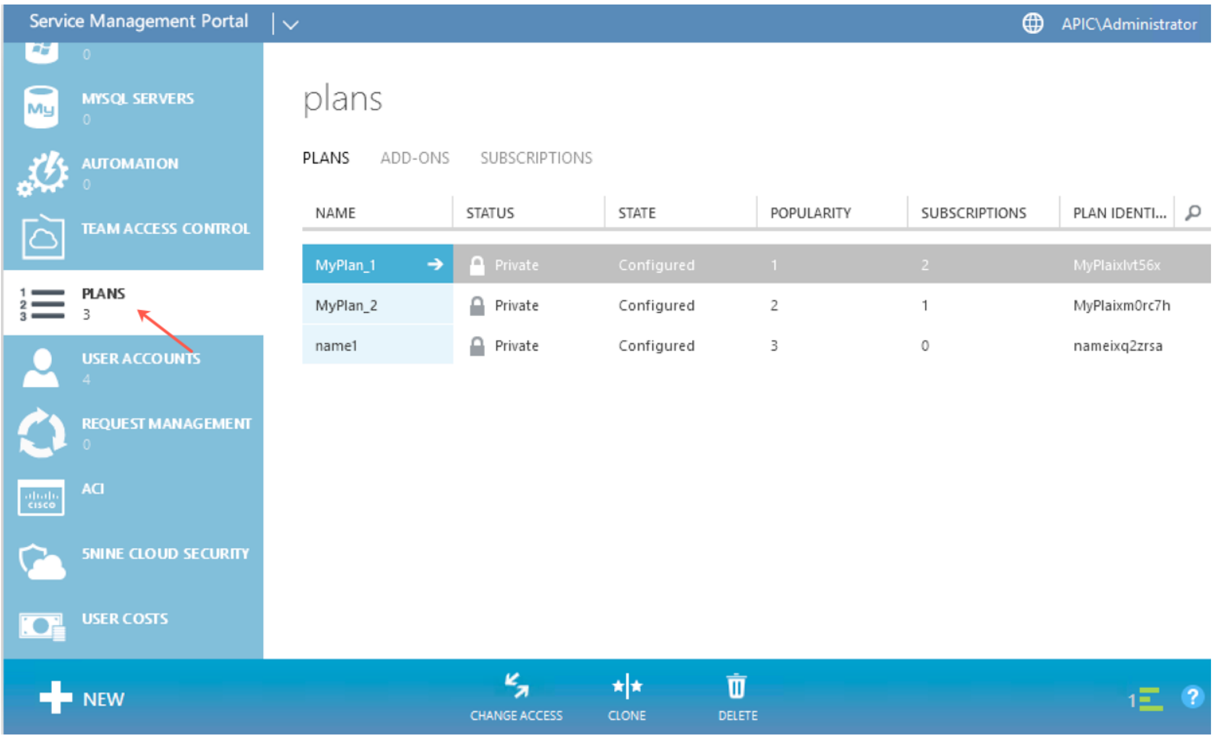
Creating a NetScaler Load Balancer in a Plan in the Service Management Portal (Admin Portal)

September 21, 2020

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

To create a NetScaler Load Balancer in a Plan in the Admin Portal:

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

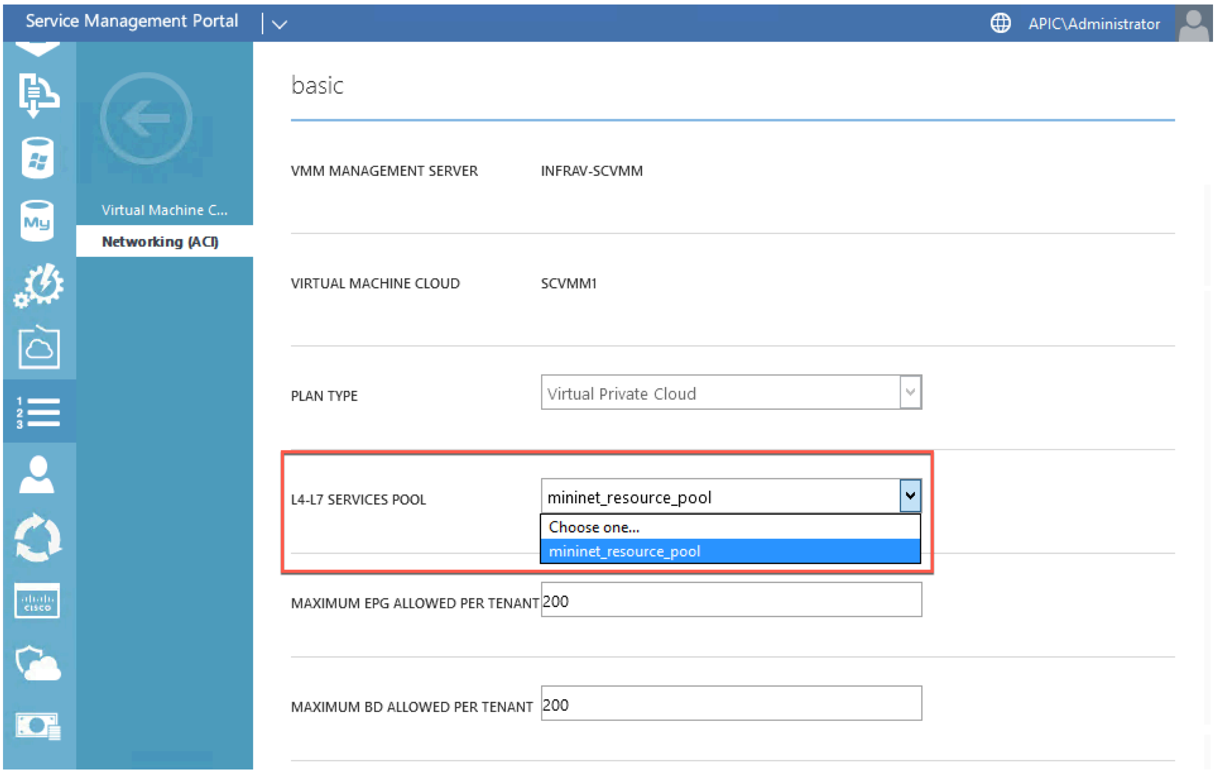

Service Management Portal
APIC\Administrator
MYSQL SERVERS
0
AUTOMATION
0
TEAM ACCESS CONTROL
PLANS
3
USER ACCOUNTS
4
REQUEST MANAGEMENT
0
ACI
SNINE CLOUD SECURITY
USER COSTS
plans
PLANS
ADD-ONS
SUBSCRIPTIONS
NEW
CHANGE ACCESS
CLONE
DELETE

NAME	STATUS	STATE	POPULARITY	SUBSCRIPTIONS	PLAN IDENTI...
MyPlan_1	Private	Configured	1	2	MyPlaixlvt56x
MyPlan_2	Private	Configured	2	1	MyPlaixm0rc7h
name1	Private	Configured	3	0	nameixq2zrsa

3. In the plans pane, select the plan that you want to add a load balancer.

4. In the selected plan's pane, select **Networking (ACI)**.

5. On the **Networking (ACI)** pane, in the **L4-L7 SERVICE POOL** drop-down list, select the L4-L7 resource pool that you had created in Cisco APIC.



VMM MANAGEMENT SERVER INFRAV-SCVMM

VIRTUAL MACHINE CLOUD SCVMM1

PLAN TYPE Virtual Private Cloud

L4-L7 SERVICES POOL mininet_resource_pool

Choose one...
mininet_resource_pool

MAXIMUM EPG ALLOWED PER TENANT 200

MAXIMUM BD ALLOWED PER TENANT 200

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

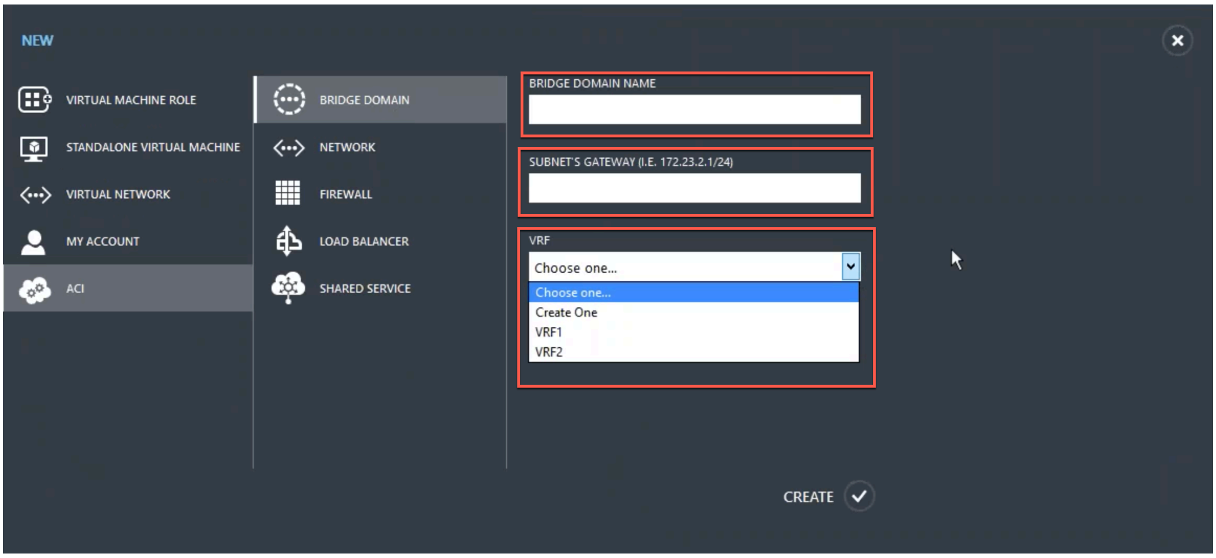
Configuring a NetScaler Load Balancer by Using the Service Management Portal (Tenant Portal)

September 21, 2020

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

To configure NetScaler Load Balancer in Service Management Portal (Tenant Portal):

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


NEW
VIRTUAL MACHINE ROLE
STANDALONE VIRTUAL MACHINE
VIRTUAL NETWORK
MY ACCOUNT
ACI
BRIDGE DOMAIN
NETWORK
FIREWALL
LOAD BALANCER
SHARED SERVICE
BRIDGE DOMAIN NAME
SUBNET'S GATEWAY (I.E. 172.23.2.1/24)
VRF
Choose one...
Choose one...
Create One
VRF1
VRF2
CREATE

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

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

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

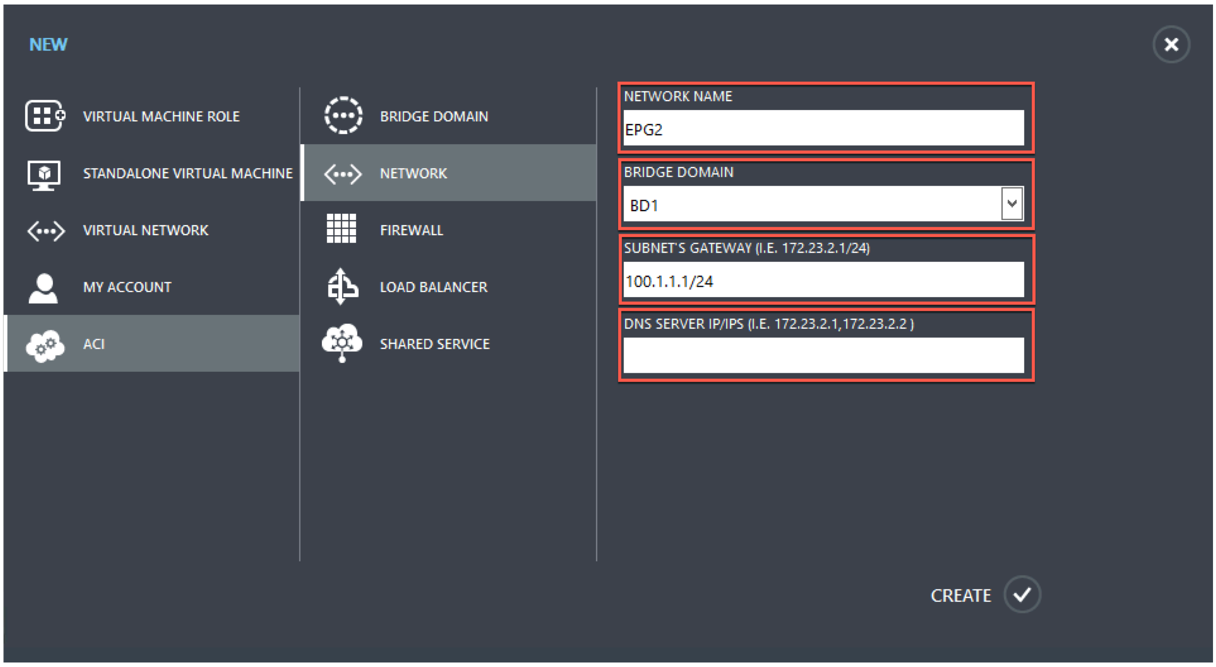
g. Click **CREATE**.

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

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

b. Click **NEW**.

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


NEW
VIRTUAL MACHINE ROLE
STANDALONE VIRTUAL MACHINE
VIRTUAL NETWORK
MY ACCOUNT
ACI
BRIDGE DOMAIN
NETWORK
FIREWALL
LOAD BALANCER
SHARED SERVICE
CREATE

NETWORK NAME: EPG2

BRIDGE DOMAIN: BD1

SUBNET'S GATEWAY (I.E. 172.23.2.1/24): 100.1.1.1/24

DNS SERVER IP/IPS (I.E. 172.23.2.1,172.23.2.2):

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

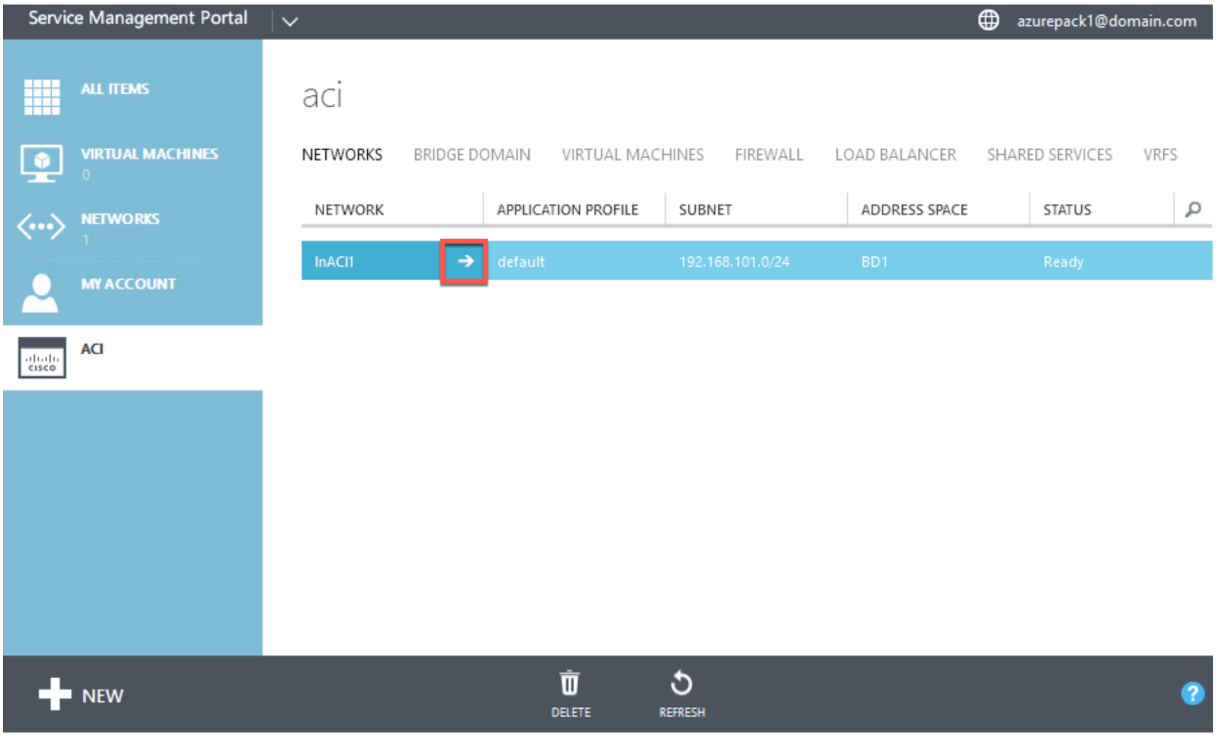
e. In the **BRIDGE DOMAIN** drop-down list, select the bridge domain you have created. (for example, BD01).

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

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

h. Click **CREATE**.

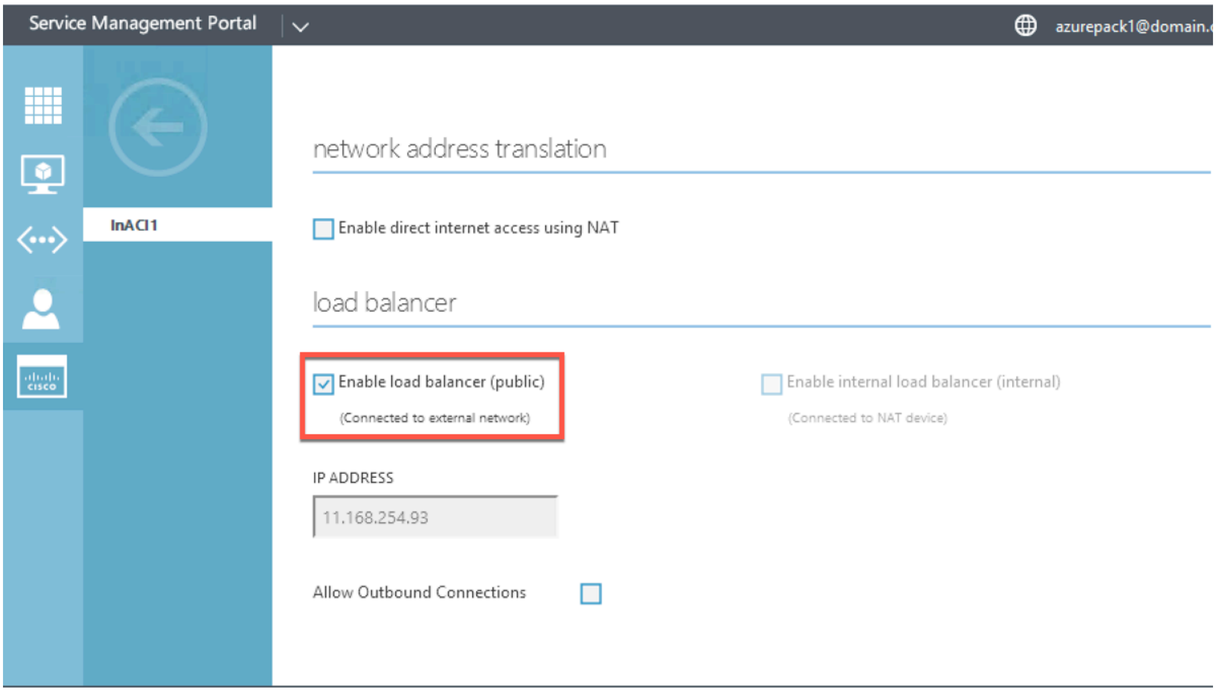
4. In the **ACI** pane, select **NETWORKS**.


Service Management Portal
azurepack1@domain.com
ALL ITEMS
VIRTUAL MACHINES
0
NETWORKS
1
MY ACCOUNT
ACI
aci
NETWORKS
BRIDGE DOMAIN
VIRTUAL MACHINES
FIREWALL
LOAD BALANCER
SHARED SERVICES
VRFS
NEW
DELETE
REFRESH

NETWORK	APPLICATION PROFILE	SUBNET	ADDRESS SPACE	STATUS
InACI1	default	192.168.101.0/24	BD1	Ready

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

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


Service Management Portal
azurepack1@domain.
InACI1
cisco

network address translation

☐ Enable direct internet access using NAT

load balancer

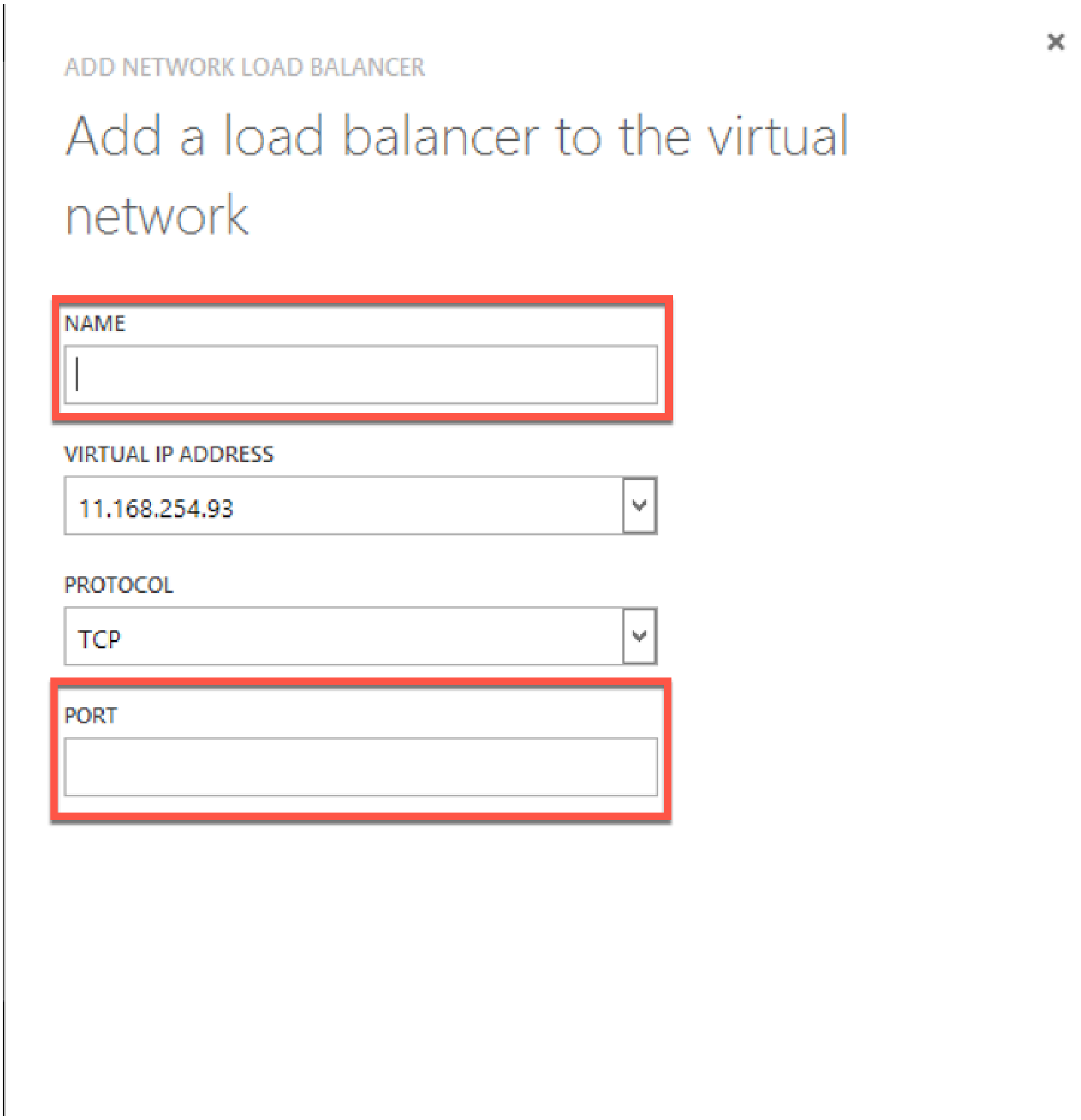
☑ Enable load balancer (public)
(Connected to external network)

☐ Enable internal load balancer (internal)
(Connected to NAT device)

IP ADDRESS
11.168.254.93

Allow Outbound Connections ☐

6. In the network pane, select the **Load Balancers** tab, and click **ADD**.


ADD NETWORK LOAD BALANCER
Add a load balancer to the virtual network
NAME
VIRTUAL IP ADDRESS
11.168.254.93
PROTOCOL
TCP
PORT

7. In the **ADD NETWORK LOAD BALANCER** pane, do the following:

   a. In the **NAME** field, enter the name for the load balancer.

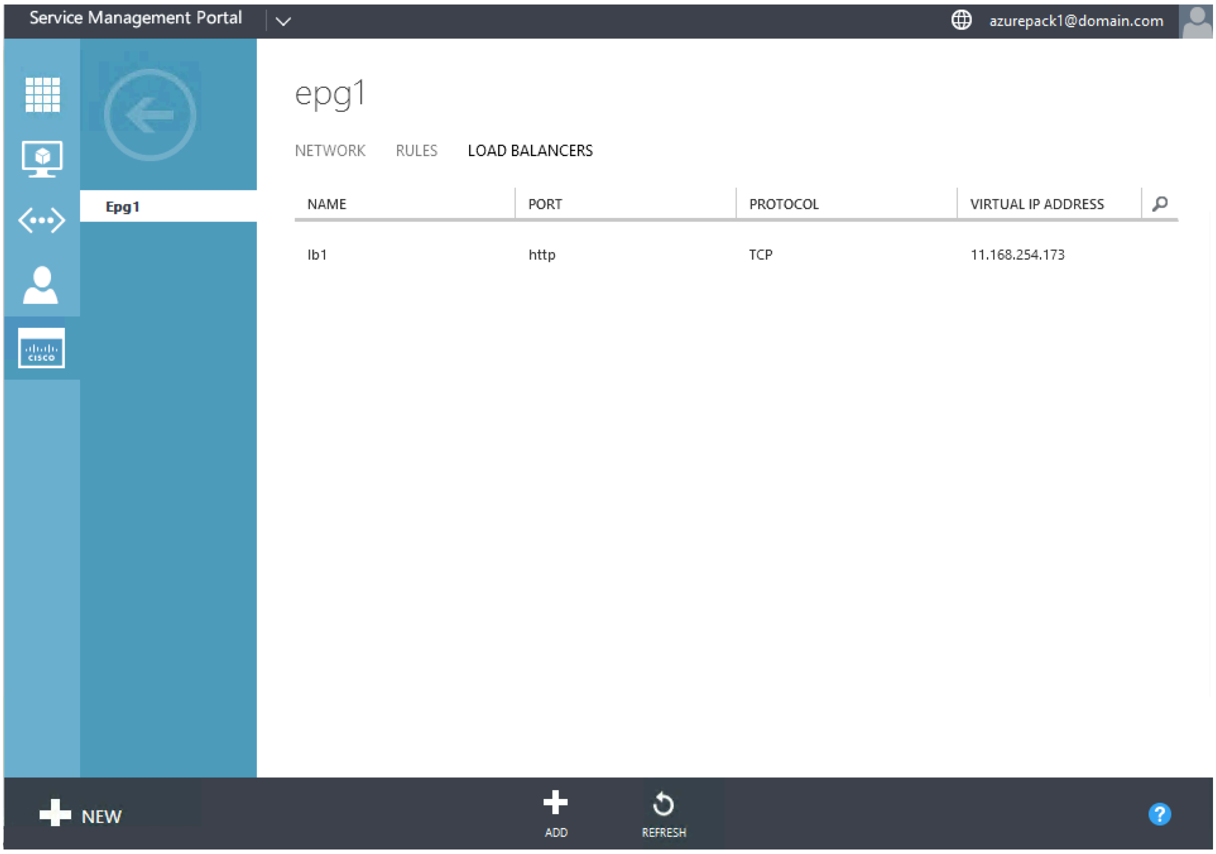
   b. Optionally, in the **VIRTUAL IP ADDRESS** field, assign the load balancer a VIP address from the VIP range that you defined earlier.

   c. Optionally, in the **PROTOCOL** field, select **TCP**.

   d. In the **PORT** field, enter the port number.

8. Click **CREATE**.

The NetScaler Load Balancer is displayed in the **LOAD BALANCERS** tab and the NetScaler Load Balancer is data path ready.


Service Management Portal
azurepack1@domain.com
Epg1
epg1
NETWORK
RULES
LOAD BALANCERS
NEW
ADD
REFRESH

NAME	PORT	PROTOCOL	VIRTUAL IP ADDRESS
lb1	http	TCP	11.168.254.173

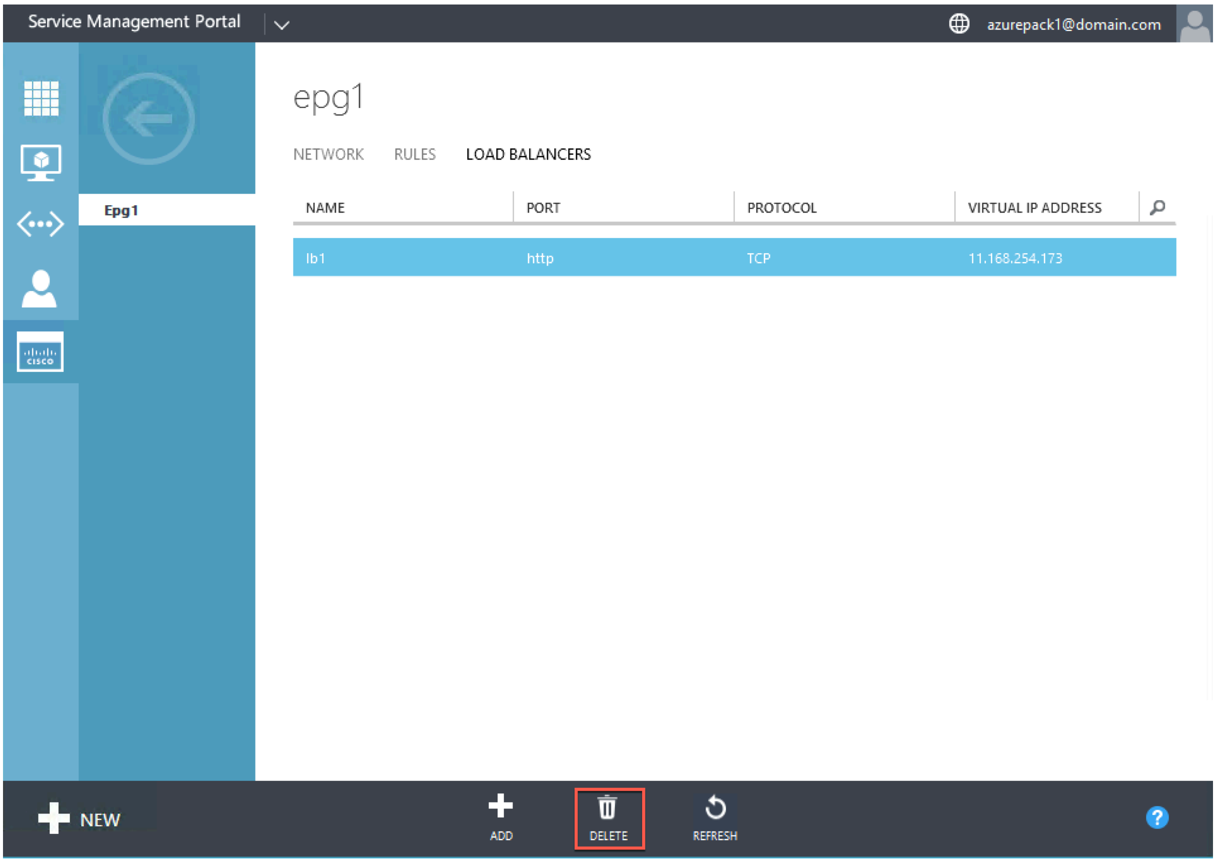
Deleting a NetScaler Load Balancer from the Network

September 21, 2020

Using Service Management Portal (Tenant Portal), from the Network, you can delete the NetScaler load balancer that you created.

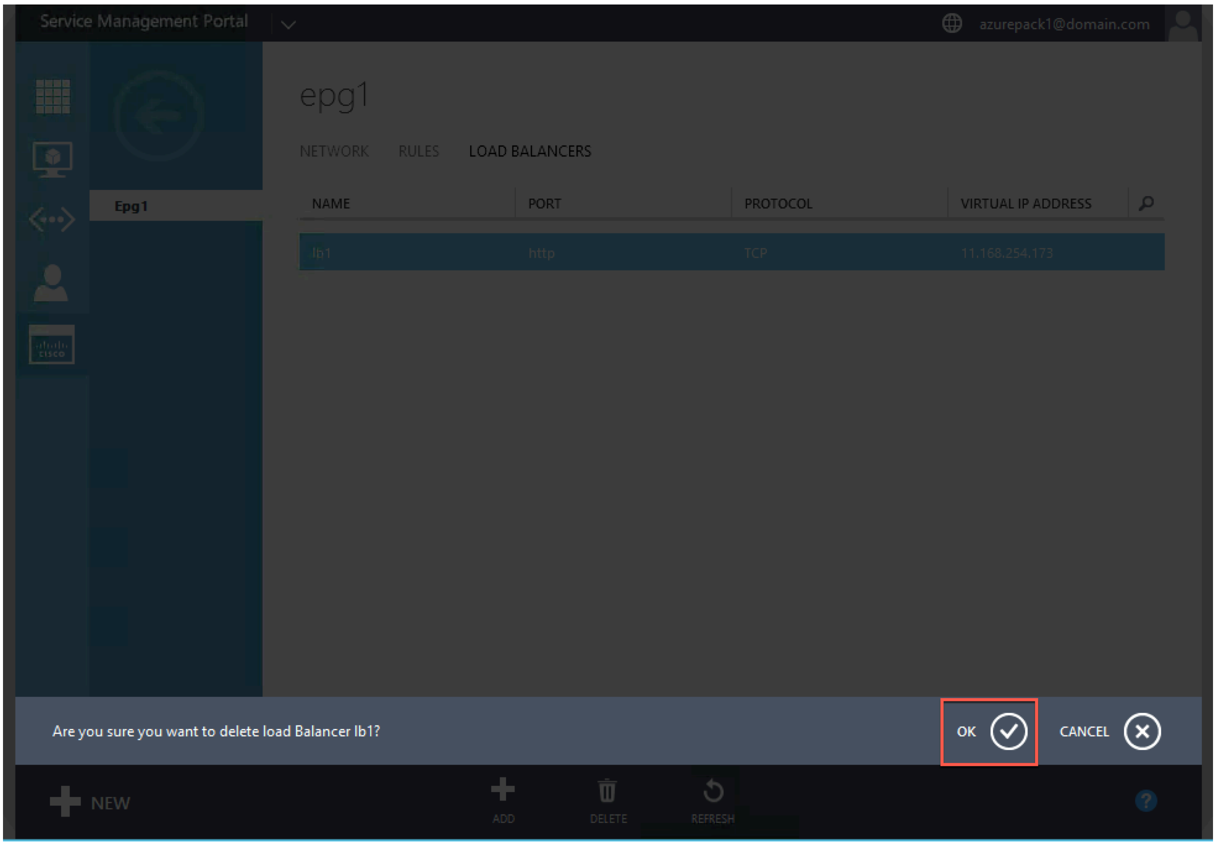
To delete a NetScaler load balancer from the Network:

1. Log on to the Service Management Portal (Tenant Portal).
2. In the navigation pane, select **ACI**.
3. In the **ACI** pane, on the **NETWORKS** tab, click the network that you created.
4. In the selected network's pane, select the NetScaler load balancer and click **DELETE**.


Service Management Portal
azurepack1@domain.com
Epg1
epg1
NETWORK
RULES
LOAD BALANCERS
NEW
ADD
DELETE
REFRESH

NAME	PORT	PROTOCOL	VIRTUAL IP ADDRESS
lb1	http	TCP	11.168.254.173

5. Click **OK** to delete the NetScaler load balancer.


Service Management Portal
azurepack1@domain.com
Epg1
epg1
NETWORK
RULES
LOAD BALANCERS
OK
CANCEL
NEW
ADD
DELETE
REFRESH

NAME	PORT	PROTOCOL	VIRTUAL IP ADDRESS
lb1	http	TCP	11.168.254.173

Are you sure you want to delete load Balancer lb1?

Deploying Citrix NetScaler VPX

September 21, 2020

The NetScaler VPX product is a virtual appliance that can be hosted on a wide variety of hypervisors and cloud platforms.

For more information, see the VPX datas sheet.

For more information about provisioning a Citrix ADC VPX instance on an SDX appliance, see Provisioning Citrix ADC instances.

Support matrix and usage guidelines

June 25, 2021

This document lists the different hypervisors and features supported on a NetScaler VPX instance, and their usage guidelines and limitations.

- Table 1-6 list the different supported hypervisors.
- Table 7 lists the different VPX features and limitations for the different hypervisors supported on a NetScaler VPX instance.
- Table 8 lists the supported web browsers that allow you access the GUI and Dashboard.

Table 1. VPX instance on XenServer

XenServer version	SysID	VPX models
6.2, 6.5	450000	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G

Table 2. VPX instance on VMware ESX server

VMware ESX version	SysID	VPX models
5.5 (build number: 3568722); 6.0 (build number: 3620759)	450010	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G, VPX 25G, VPX 40G, VPX 100G

Table 3. VPX on Microsoft Hyper-V

Hyper-V version	SysID	VPX models
2012, 2012R2	450020	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000

Table 4. VPX instance on generic KVM

Generic KVM version	SysID	VPX models
RHEL 7.2, Ubuntu 15	450070	VPX 10, VPX 25, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 8000, VPX 10G, VPX 15G. VPX 25G, VPX 40G, VPX 100G

> Note: The VPX instance is qualified for hypervisor release versions mentioned in table 1–4, and not for patch releases within a version. However, the VPX instance is expected to work seamlessly with patch releases of a supported version. If it does not, log a support case for troubleshooting and debugging.

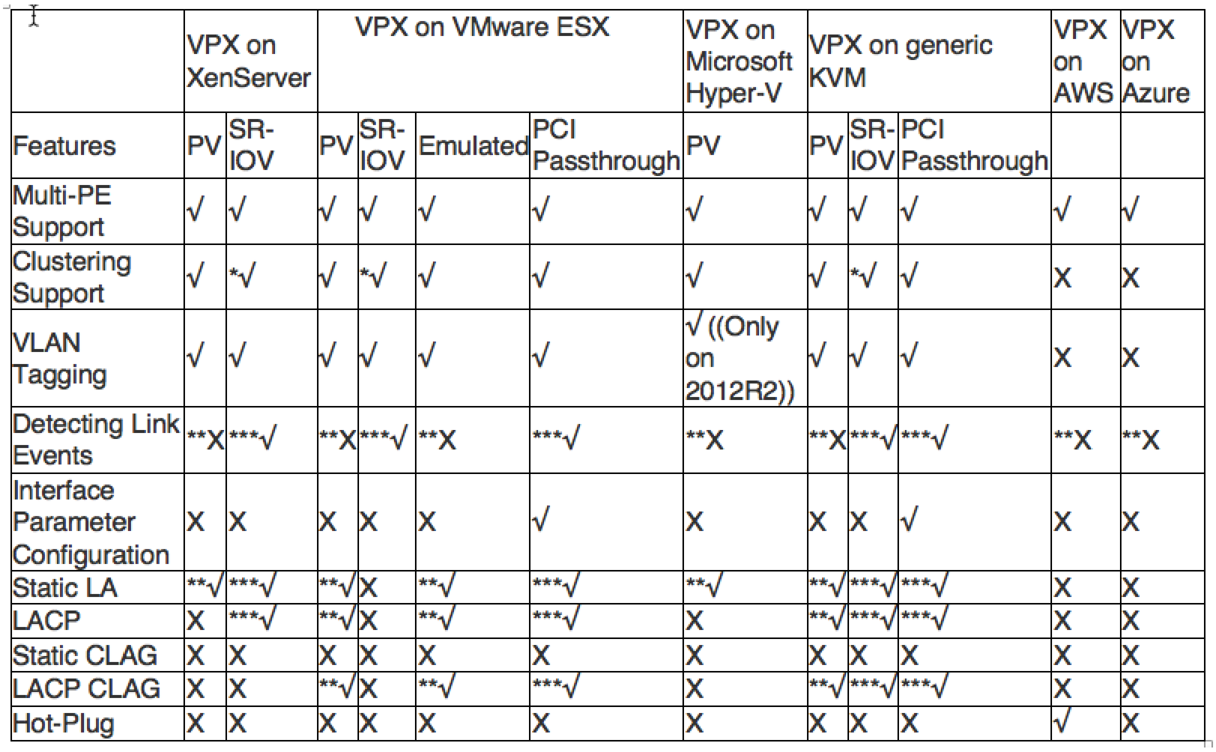
Table 5. VPX instance on AWS

AWS version	SysID	VPX models
N/A	450040	VPX 10, VPX 200, VPX 1000, VPX 3000, VPX 5000, VPX 15G, VPX BYOL

Table 6. VPX instance on Azure

Azure version	SysID	VPX models
N/A	450020	VPX 10, VPX 200, VPX 1000, VPX 3000, VPX BYOL

Table 7. VPX feature matrix



	VPX on XenServer		VPX on VMware ESX				VPX on Microsoft Hyper-V	VPX on generic KVM			VPX on AWS	VPX on Azure
Features	PV	SR-IOV	PV	SR-IOV	Emulated	PCI Passthrough	PV	PV	SR-IOV	PCI Passthrough		
Multi-PE Support	√	√	√	√	√	√	√	√	√	√	√	√
Clustering Support	√	*√	√	*√	√	√	√	√	*√	√	X	X
VLAN Tagging	√	√	√	√	√	√	√ ((Only on 2012R2))	√	√	√	X	X
Detecting Link Events	**X	***√	**X	***√	**X	***√	**X	**X	***√	***√	**X	**X
Interface Parameter Configuration	X	X	X	X	X	√	X	X	X	√	X	X
Static LA	**√	***√	**√	X	**√	***√	**√	**√	***√	***√	X	X
LACP	X	***√	**√	X	**√	***√	X	**√	***√	***√	X	X
Static CLAG	X	X	X	X	X	X	X	X	X	X	X	X
LACP CLAG	X	X	**√	X	**√	***√	X	**√	***√	***√	X	X
Hot-Plug	X	X	X	X	X	X	X	X	X	X	√	X

*Clustering support is available on SRIOV for client- and server-facing interfaces and not for the backplane.

**Interface DOWN events are not recorded in NetScaler VPX instances.

For Static LA, traffic might still be sent on the interface whose physical status is DOWN.

For LACP, peer device knows interface DOWN event based on LACP timeout mechanism.

Short timeout: 3 seconds

Long timeout: 90 seconds

For LACP, interfaces must not be shared across VMs.

For Dynamic routing, convergence time depends on the Routing Protocol since link events are not detected.

Monitored static Route functionality fails if monitors are not bound to static routes since Route state depends on the VLAN status. The VLAN status depends on the link status.

Partial failure detection does not happen in high availability if there is link failure. High availability-split brain condition might happen if there is link failure.

***When any link event (disable/enable, reset) is generated from a VPX instance, the physical status of the link does not change. For static LA, any traffic initiated by the peer gets dropped on the instance.

For LACP, peer device knows interface DOWN event based on LACP timeout mechanism.

Short timeout: 3 seconds

Long timeout: 90 seconds

For LACP, interfaces should not be shared across VMs.

- For the VLAN tagging feature to work, do the following:

On the VMware ESX, set the port group's VLAN ID to 1 - 4095 on the vSwitch of VMware ESX server. For more information about setting a VLAN ID on the vSwitch of VMware ESX server, see pdf.

Table 8. Supported browsers

Operating system	Browser and versions
Windows 7	Internet Explorer- 8, 9, 10, and 11; Mozilla Firefox 3.6.25 and above; Google Chrome- 15 and above
Windows 64 bit	Internet Explorer - 8, 9; Google Chrome - 15 and above
MAC	Mozilla Firefox - 12 and above; Safari - 5.1.3; Google Chrome - 15 and above

Usage guidelines

Follow these usage guidelines:

- See the **VMware ESXi CPU Considerations** section in the document Performance Best Practices for VMware vSphere 6.5. Here's an extract:

  It is not recommended that virtual machines with high CPU/Memory demand sit on a Host/Cluster that is overcommitted.
  In most environments ESXi allows significant levels of CPU overcommitment (that is, running more
  vCPUs on a host than the total number of physical processor cores in that host) without impacting virtual
  machine performance.
  If an ESXi host becomes CPU saturated (that is, the virtual machines and other loads on the host demand
  all the CPU resources the host has), latency-sensitive workloads might not perform well. In this case you
  might want to reduce the CPU load, for example by powering off some virtual machines or migrating them to a different host (or allowing DRS to migrate them automatically).

- The Citrix ADC VPX is a latency-sensitive, high-performance virtual appliance. To deliver its expected performance, the appliance requires vCPU reservation, memory reservation, vCPU pinning on the host. Also, hyper threading must be disabled on the host. If the host does not meet these requirements, issues such as high-availability failover, CPU spike within the VPX instance, sluggishness in accessing the VPX CLI, pitboss daemon crash, packet drops, and low throughput occur.

- A hypervisor is considered over-provisioned if one of the following two conditions is met:
  - The total number of virtual cores (vCPU) provisioned on the host is greater than the total number of physical cores (pCPUs).
  - The total number of provisioned VMs consume more vCPUs than the total number of pCPUs.

    At times, if an instance is over-provisioned, the hypervisor might not be able to guarantee the resources reserved (such as CPU, memory, and others) for the instance due to hypervisor scheduling over-heads or bugs or limitations with the hypervisor. This can cause lack of CPU resource for Citrix ADC and might lead to issues mentioned in the first point under **Usage guidelines**. As administrators, you're recommended to reduce the tenancy on the host so that the total number of vCPUs provisioned on the host is lesser or equal to the total number of pCPUs.

    Example

For ESX hypervisor, if the `%RDY%` parameter of a VPX vCPU is greater than 0 in the `esxtop` command output, the ESX host is said to be having scheduling overheads, which can cause latency related issues for the VPX instance.

In such a situation, reduce the tenancy on the host so that `%RDY%` returns to 0 always. Alternatively, contact the hypervisor vendor to triage the reason for not honoring the resource reservation done.

Install a NetScaler VPX instance on XenServer

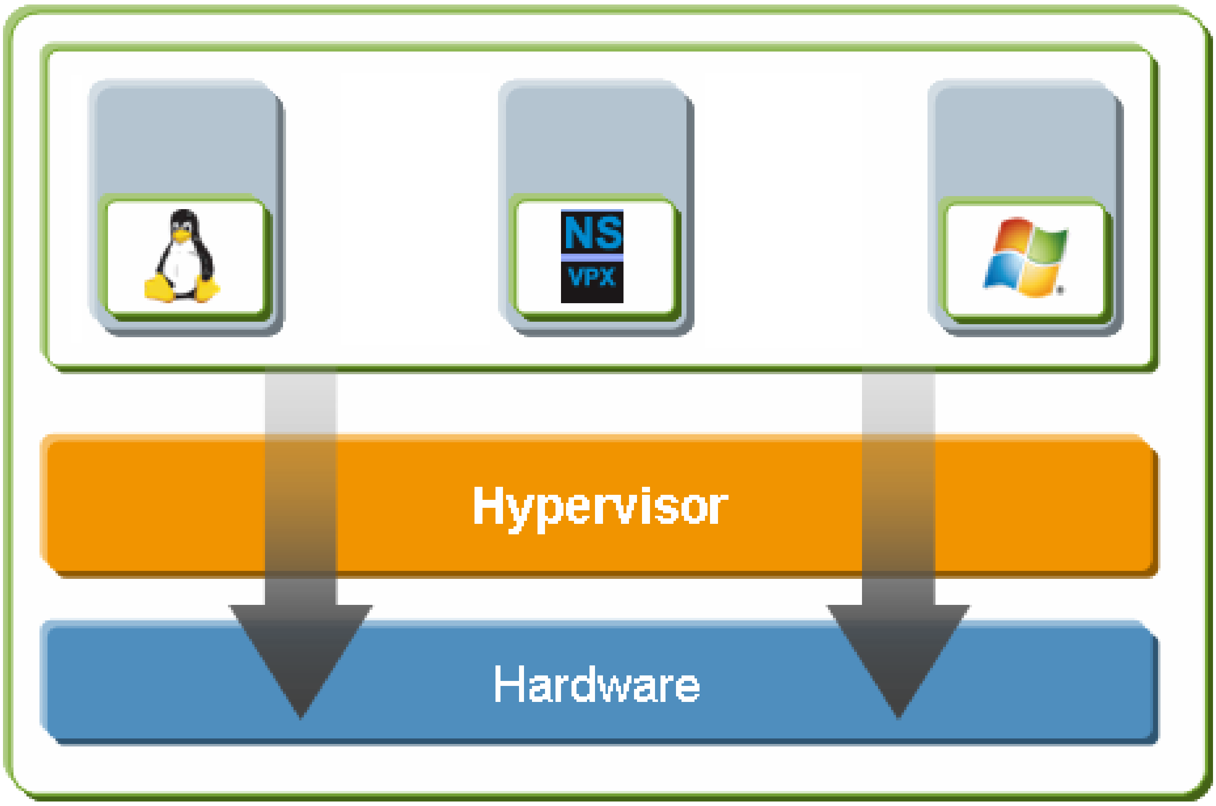
September 21, 2020

To install VPX instances on Citrix XenServer, you must first install XenServer on a machine with adequate system resources. To perform the NetScaler VPX instance installation, you use Citrix XenCenter, which must be installed on a remote machine that can connect to the XenServer host through the network.

For more information about XenServer, see XenServer documentation.

The following figure shows the bare-metal solution architecture of NetScaler VPX instance on XenServer.

Figure. A NetScaler VPX instance on XenServer



Prerequisites for installing a NetScaler VPX instance on XenServer

Before you begin installing a virtual appliance, do the following:

- Install XenServer version 6.0 or later on hardware that meets the minimum requirements.
- Install XenCenter on a management workstation that meets the minimum system requirements.
- Obtain virtual appliance license files. For more information about virtual appliance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx122426.

XenServer hardware requirements

The following table describes the minimum hardware requirements for a XenServer platform running a NetScaler VPX instance.

Table 1. Minimum system requirements for XenServer running a nCore VPX instance

Component	Requirement
CPU	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. AMD processor is not supported. To run NetScaler VPX instance, hardware support for virtualization must be enabled on the XenServer host. Make sure that the BIOS option for virtualization support is not disabled. For more details, see BIOS documentation.
RAM	3 GB
Disk space	Locally attached storage (PATA, SATA, SCSI) with 40 GB of disk space. Note: XenServer installation creates a 4 GB partition for the XenServer host control domain; the remaining space is available for NetScaler VPX instance and other virtual machines.
NIC	One 1-Gbps NIC; recommended: two 1-Gbps NICs

For information about installing XenServer, see the XenServer documentation at http://support.citrix.com/product/xens/.

The following table lists the virtual computing resources that XenServer must provide for each nCore VPX virtual appliance.

Table 2. Minimum virtual computing resources required for running a ncore VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	2

> Note: For production use of
> NetScaler VPX instance, Citrix recommends that CPU priority (in virtual machine properties) be set to the highest level, in order to improve scheduling behavior and network latency.

XenCenter system requirements

XenCenter is a Windows client application. It cannot run on the same machine as the XenServer host. For more information about minimum system requirements and installing XenCenter, see the following XenServer documents:

- System requirements
- Install

Install NetScaler VPX instances on XenServer by using XenCenter

After you have installed and configured XenServer and XenCenter, you can use XenCenter to install virtual appliances on XenServer. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running XenServer.

After you have used XenCenter to install the initial NetScaler VPX instance (.xva image) on XenServer, you have the option to use Command Center to provision NetScaler VPX instance. For more information, see the Command Center documentation.

To install NetScaler VPX instances on XenServer by using XenCenter, follow these steps:

1. Start XenCenter on your workstation.
2. On the Server menu, click Add.
3. In the Add New Server dialog box, in the Hostname text box, type the IP address or DNS name of the XenServer that you want to connect to.
4. In the User Name and Password text boxes, type the administrator credentials, and then click Connect. The XenServer name appears in the navigation pane with a green circle, which indicates that the XenServer is connected.
5. In the navigation pane, click the name of the XenServer on which you want to install NetScaler VPX instance.

6. On the VM menu, click Import.
7. In the Import dialog box, in Import file name, browse to the location at which you saved the NetScaler VPX instance .xva image file. Make sure that the Exported VM option is selected, and then click Next.
8. Select the XenServer on which you want to install the virtual appliance, and then click Next.
9. Select the local storage repository in which to store the virtual appliance, and then click Import to begin the import process.
10. You can add, modify, or delete virtual network interfaces as required. When finished, click Next.
11. Click Finish to complete the import process.
    Note: To view the status of the import process, click the
    Log tab.
12. If you want to install another virtual appliance, repeat steps 5 through 11.

> **Note**
>
> After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrading or Downgrading the System Software.

Configure VPX instances to use single root I/O virtualization (SR-IOV) network interfaces

September 21, 2020

After you have installed and configured a NetScaler VPX instance on XenServer, you can configure the virtual appliance to use SR-IOV network interfaces.

Limitations

XenServer does not support the following features on SRIOV interfaces:

- L2 mode switching
- Clustering
- Admin partitioning [Shared VLAN mode]
- High Availability [Active - Active mode]
- Jumbo frames
- IPv6 protocol in Cluster environment

Prerequisites

On the XenServer host, ensure that you:

- Add the Intel 82599 Network Interface Card (NIC) to the host.
- Blacklist the ixgbevf driver by adding the following entry to the **/etc/modprobe.d/blacklist.conf** file:
  blacklist ixgbevf
- Enable SR-IOV Virtual Functions (VFs) by adding the following entry to the **/etc/modprobe.d/ixgbe** file:
  options ixgbe max_vfs=<*number_of_VFs*>
  where <*number_VFs*> is the number of SR-IOV VFs that you want to create.
- Verify that SR-IOV is enabled in BIOS.

IXGBE driver version 3.22.3 is recommended.

Assign SR-IOV VFs to the VPX instance by using the XenServer host

To assign SR-IOV network interfaces to NetScaler VPX instance, follow these steps:

1. On the XenServer host, use the following command to assign the SR-IOV VFs to the NetScaler VPX instance:

xe host-call-plugin plugin=iovirt host-uuid=<*Xen host UUID*> **fn=assign_free_vf args:uuid**=<*Netscalar VM UUID*> **args:ethdev**=<*interface name*> **args:mac=**<*mac addr*>

Where:

• <*Xen host UUID*> is the UUID of the XenServer host.

• <*Netscalar VM UUID*> is the UUID of the NetScaler VPX instance.

• <*interface name*> is the interface for the SR-IOV VFs.

• <*mac addr* > is the mac address of the SR-IOV VF.

> **Note**
>
> Specify the mac address that you want use in the args:mac= parameter, if not specified, the iovirt script randomly generates and assigns a mac address. Also, if you want use the SR-IOV VFs in Link Aggregation mode, make sure that you specify the mac address as 00:00:00:00:00:00.

2. Boot the NetScaler VPX instance.

Unassign SR-IOV VFs to the VPX instance by using the XenServer host

If you have assigned an incorrect SR-IOV VFs or if you want modify the a assigned SR-IOV VFs, you need to unassign and reassign the SR-IOV VFs to the NetScaler VPX instance.

To unassign SR-IOV network interface assigned to a NetScaler VPX instance, follow these steps:

1. On the XenServer host, use the following command to assign the SR-IOV VFs to the NetScaler VPX instance and reboot the NetScaler VPX instance:

```
xe host-call-plugin plugin=iovirt host-uuid=<Xen_host_UUID> fn=unassign_all args:uuid=<Netscalar_VM_UUID
```

Where:

- *<Xen_host_UUID>* - The UUID of the XenServer host.
- *<Netscalar_VM_UUID>* - The UUID of the NetScaler VPX instance

2. Boot the NetScaler VPX instance.

Configuring VLAN on the SR-IOV Interface

> **Important**
>
> While you are assigning the SR-IOV VFs to the NetScaler VPX instance, make sure that you specify MAC address 00:00:00:00:00:00 for the VFs.

To use the SR-IOV virtual functions in link aggregation mode, you need to disable spoof checking for virtual functions that you have created. On the XenServer host, use the following command to disable spoof checking:

```
ip link set <interface_name> vf <VF_id> spoofchk off
```

Where:

- <interface_name> is the interface name.
- <VF_id> is the virtual function ID.

After disabling spoof checking for all the Virtual Function that you have created, restart the NetScaler VPX instance and configure link aggregation. For instructions, see Configure link aggregation.

Configure VLAN on the SR-IOV interface

You can configure VLAN on the SR-IOV Virtual Functions, for instructions, see Configuring a VLAN.

> **Important**
>
> Make sure that the XenServer host does not contain VLAN settings for the VF interface.

Install a NetScaler VPX instance on VMware ESX

September 21, 2020

Before installing
NetScaler VPX instances on VMware ESX, make sure that VMware ESX Server is installed on a machine

with adequate system resources. To install a NetScaler VPX instance on VMware ESXi
, you use VMware vSphere client. The client or tool must be installed on a remote machine that can connect to VMware ESX through the network.

> **Important**
>
> You cannot install standard VMware Tools or upgrade the VMware Tools version available on a NetScaler VPX instance. VMware Tools for a NetScaler VPX instance are delivered as part of the NetScaler software release.

Prerequisites

Before you begin installing a virtual appliance, do the following:

- Install VMware ESX on hardware that meets the minimum requirements.
- Install VMware Client on a management workstation that meets the minimum system requirements.
- Download the NetScaler VPX appliance setup files.
- Label the physical network ports of VMware ESX.
- Obtain VPX license files. For more information about NetScaler VPX instance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

VMware ESX hardware requirements

The following table describes the minimum system requirements for VMware ESX servers running NetScaler VPX ncore virtual appliance.

Table 1. Minimum system requirements for a VMware ESX server running a NetScaler VPX instance

Component	Requirement
-	2 or more 64-bit x86 CPUs with virtualization assist (Intel-VT) enabled. Note that to run NetScaler VPX instance, hardware support for virtualization must be enabled on the VMware ESX host. Make sure that the BIOS option for virtualization support is not disabled. For more information, see your BIOS documentation.
RAM	3 GB
Disk space	40 GB of disk space available
Network	One 1-Gbps NIC; Two 1-Gbps NICs recommended

For information about installing VMware ESX, see http://www.vmware.com/.

The following table lists the virtual computing resources that the VMware ESX server must provide for each VPX ncore virtual appliance.

Table 2. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
Memory	2 GB
Virtual CPU (VCPU)	2
Virtual network interfaces	1. Note that with ESX, you can install a maximum of 10 virtual network interfaces if the VPX hardware is upgraded version to 7 or higher.
Disk space	20 GB

> Note that this is in addition to any disk requirements for the hypervisor.

For production use of VPX virtual appliance, the full memory allocation must be reserved. CPU cycles (in MHz) equal to at least the speed of one CPU core of the ESX should also be reserved.

VMware vSphere client system requirements

VMware vSphere is a client application that can run on Windows and Linux operating systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 3. Minimum system requirements for VMware vSphere client installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "vSphere Compatibility Matrixes" PDF file at http://kb.vmware.com/.
CPU	750 megahertz (MHz); 1 gigahertz (GHz) or faster recommended
RAM	1 GB; 2 GB recommended
Network Interface Card (NIC)	100 Mbps or faster NIC

OVF Tool 1.0 system requirements

OVF Tool is a client application that can run on Windows and Linux systems. It cannot run on the same machine as the VMware ESX server. The following table describes the minimum system requirements.

Table 4. Minimum system requirements for OVF tool installation

Component	Requirement
Operating system	For detailed requirements from VMware, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.
CPU	750 MHz minimum, 1 GHz or faster recommended
RAM	1 GB Minimum, 2 GB recommended
Network Interface Card (NIC)	100 Mbps or faster NIC

For information about installing OVF, search for the "OVF Tool User Guide" PDF file at http://kb.vmware.com/.

Downloading the NetScaler VPX setup files

The NetScaler VPX instance setup package for VMware ESX follows the Open Virtual Machine (OVF) format standard. You can download the files from the Citrix website. You need a Citrix account to log on. If you do not have a Citrix account, access the home page at http://www.citrix.com, click the New Users link, and follow the instructions to create a new Citrix account.

Once logged on, navigate the following path from the Citrix home page:

Citrix.com > Downloads > NetScaler > Virtual Appliances.

Copy the following files to a workstation on the same network as the ESX server. Copy all three files into the same folder.

- NSVPX-ESX-<release number>-<build number>-disk1.vmdk (for example, NSVPX-ESX-9.3-39.8-disk1.vmdk)
- NSVPX-ESX-<release number>-<build number>.ovf (for example, NSVPX-ESX-9.3-39.8.ovf)
- NSVPX-ESX-<release number>-<build number>.mf (for example, NSVPX-ESX-9.3-39.8.mf)

Label the physical network ports of VMware ESX

Before installing a VPX virtual appliance, label of all the interfaces that you plan to assign to virtual appliances, in a unique format, for example, NS_NIC_1_1, NS_NIC_1_2, and so on. In large deployments, labeling in a unique format helps in quickly identifying the interfaces that are allocated to the VPX virtual appliance among other interfaces used by other virtual machines, such as Windows and Linux. Such labeling is especially important when different types of virtual machines share the same interfaces.

To label the physical network ports of VMware ESX server, follow these steps:

1. Log on to the VMware ESX server by using the vSphere client.
2. On the vSphere client, select the Configuration tab, and then click Networking.
3. At the top-right corner, click Add Networking.
4. In the Add Network Wizard, for **Connection Type**, select **Virtual Machine**, and then click Next.
5. Scroll through the list of vSwitch physical adapters, and choose the physical port that will map to interface 1/1 on the virtual appliances.
6. Enter the label of the interface, for example, **NS_NIC_1_1** as the name of the vSwitch that will be associated with interface 1/1 of the virtual appliances.
7. Click Next to finish the vSwitch creation. Repeat the procedure, beginning with step 2, to add any additional interfaces to be used by your virtual appliances. Label the interfaces sequentially, in the correct format (for example, NS_NIC_1_2).

Install a NetScaler VPX instance on VMware ESX

After you have installed and configured VMware ESX, you can use the VMware vSphere client to install virtual appliances on the VMware ESX server. The number of virtual appliances that you can install depends on the amount of memory available on the hardware that is running VMware ESX.

To install NetScaler VPX instances on VMware ESX by using VMware vSphere Client, follow these steps:

1. Start the VMware vSphere client on your workstation.
2. In the **IP address / Name** text box, type the IP address of the VMware ESX server that you want to connect to.
3. In the **User Name** and **Password** text boxes, type the administrator credentials, and then click Login.
4. On the **File** menu, click **Deploy OVF Template**.
5. In the **Deploy OVF Template** dialog box, in **Deploy from file**, browse to the location at which you saved the NetScaler VPX instance setup files, select the .ovf file, and click **Next**.
6. Map the networks shown in the virtual appliance OVF template to the networks that you configured on the ESX host. Click **Next**to start installing a virtual appliance on VMware ESX. When installation is complete, a pop-up window informs you of the successful installation.
7. You are now ready to start the NetScaler VPX instance. In the navigation pane, select the NetScaler VPX instance that you have just installed and, from the right-click menu, select **Power On**. Click the **Console**tab to emulate a console port.
8. If you want to install another virtual appliance, repeat steps **4** through **6**.

> **Note**
>
> By default, the NetScaler VPX instance uses E1000 network interfaces.

After the installation, you can use vSphere client or vSphere Web Client to manage virtual appliances on VMware ESX.

> **Note**
>
> For the VLAN tagging feature to work,on the VMware ESX, set the port group's VLAN ID to 1 - 4095 on the VSwitch of VMware ESX server. For more information about setting a VLAN ID on the VSwitch of VMware ESX server, see http://www.vmware.com/pdf/esx3_vlan_wp.pdf.

Configure a NetScaler VPX instance to use VMXNET3 network interface

September 21, 2020

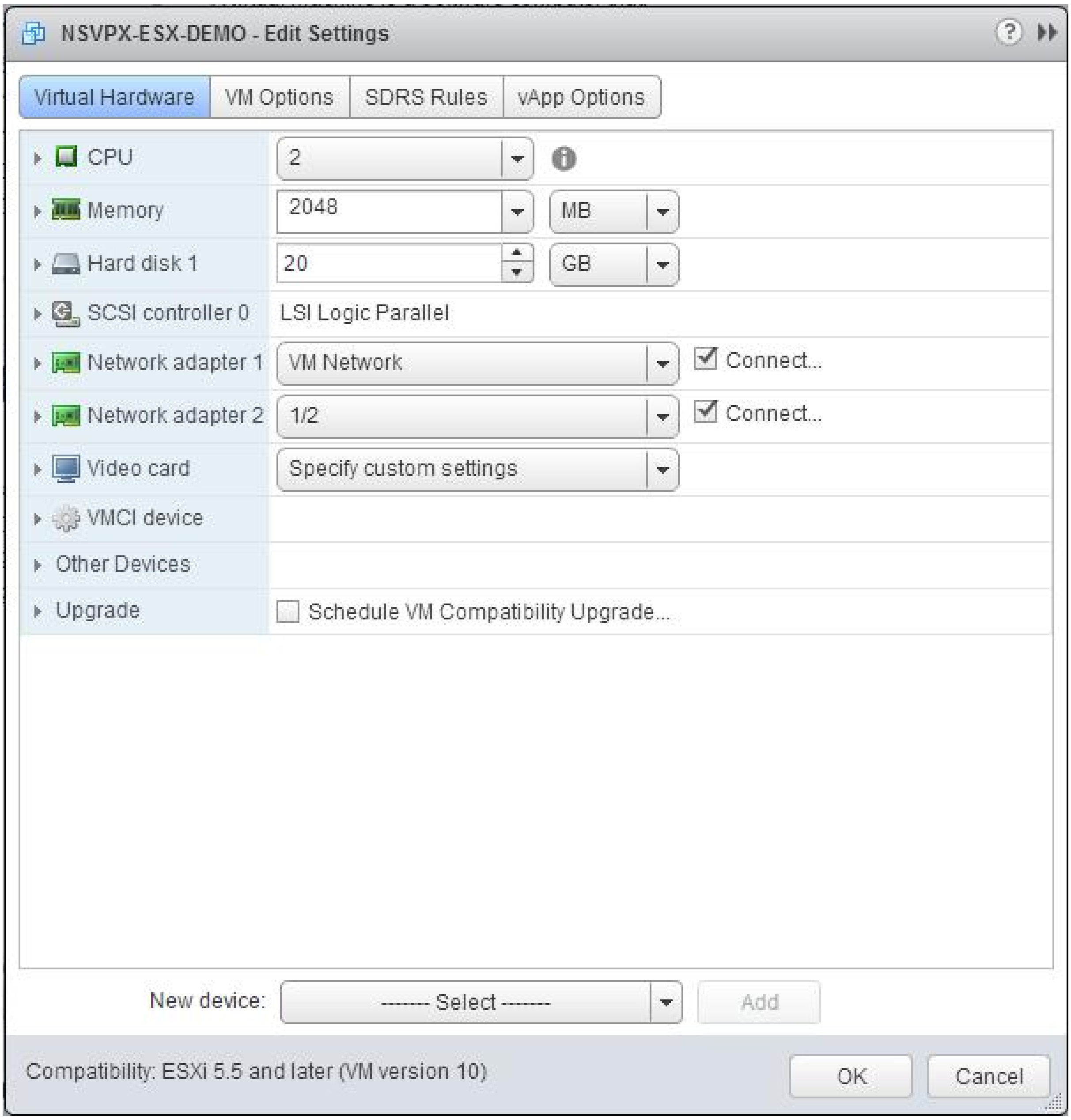
After you have installed and configured the NetScaler VPX instance on the VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use VMXNET3 network interfaces.

To configure NetScaler VPX instances to use VMXNET3 network interfaces by using the VMware vSphere Web Client:

1. In the vSphere Web Client, select Hosts and Clusters.
2. Upgrade the Compatibility setting of the NetScaler VPX instance to ESX, as follows:
   a. Power off the NetScaler VPX instance.
   b. Right-click the NetScaler VPX instance and select Compatibility > Upgrade VM Compatibility.

c. In the Configure VM Compatibility dialog box, select ESXi 5.5 and later from the Compatible with drop-down list and click OK.

3. Right-click on the NetScaler VPX instance and click Edit Settings.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

CPU: 2

Memory: 2048 MB

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Network adapter 2: 1/2 ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

Upgrade: ☐ Schedule VM Compatibility Upgrade...

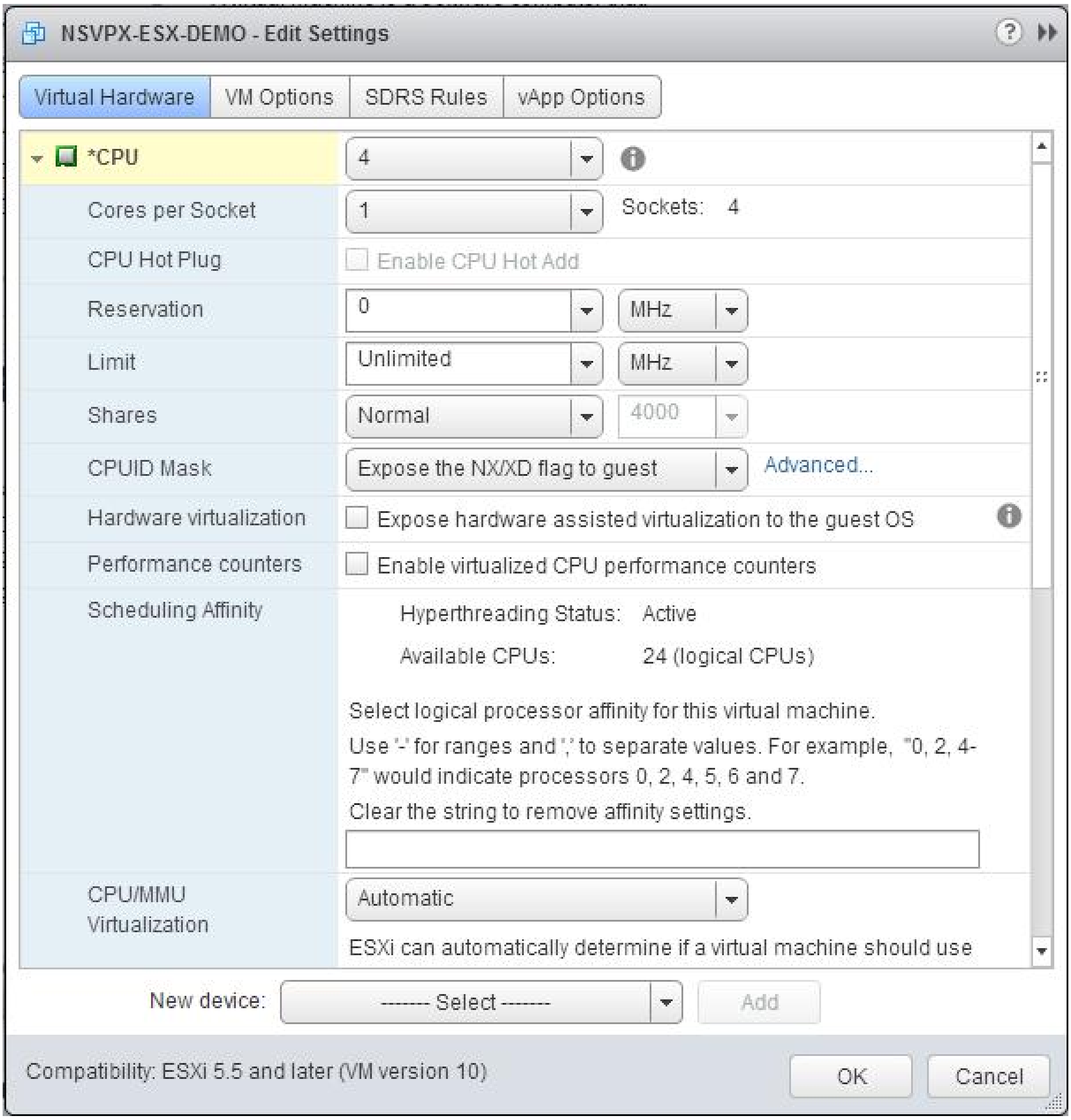
New device: ------- Select -------

Add

Compatibility: ESXi 5.5 and later (VM version 10)

OK
Cancel

4. In the <virtual_appliance> - Edit Settings dialog box, click the CPU section.



*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation: 0 MHz

Limit: Unlimited MHz

Shares: Normal 4000

CPUID Mask: Expose the NX/XD flag to guest

Hardware virtualization: ☐ Expose hardware assisted virtualization to the guest OS

Performance counters: ☐ Enable virtualized CPU performance counters

Scheduling Affinity:

Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

Select logical processor affinity for this virtual machine.

Use '-' for ranges and ',' to separate values. For example, "0, 2, 4-7" would indicate processors 0, 2, 4, 5, 6 and 7.

Clear the string to remove affinity settings.

CPU/MMU Virtualization: Automatic

ESXi can automatically determine if a virtual machine should use

New device: ------- Select -------

Compatibility: ESXi 5.5 and later (VM version 10)

5. In the CPU section, update the following:

   - Number of CPUs
   - Number of Sockets
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

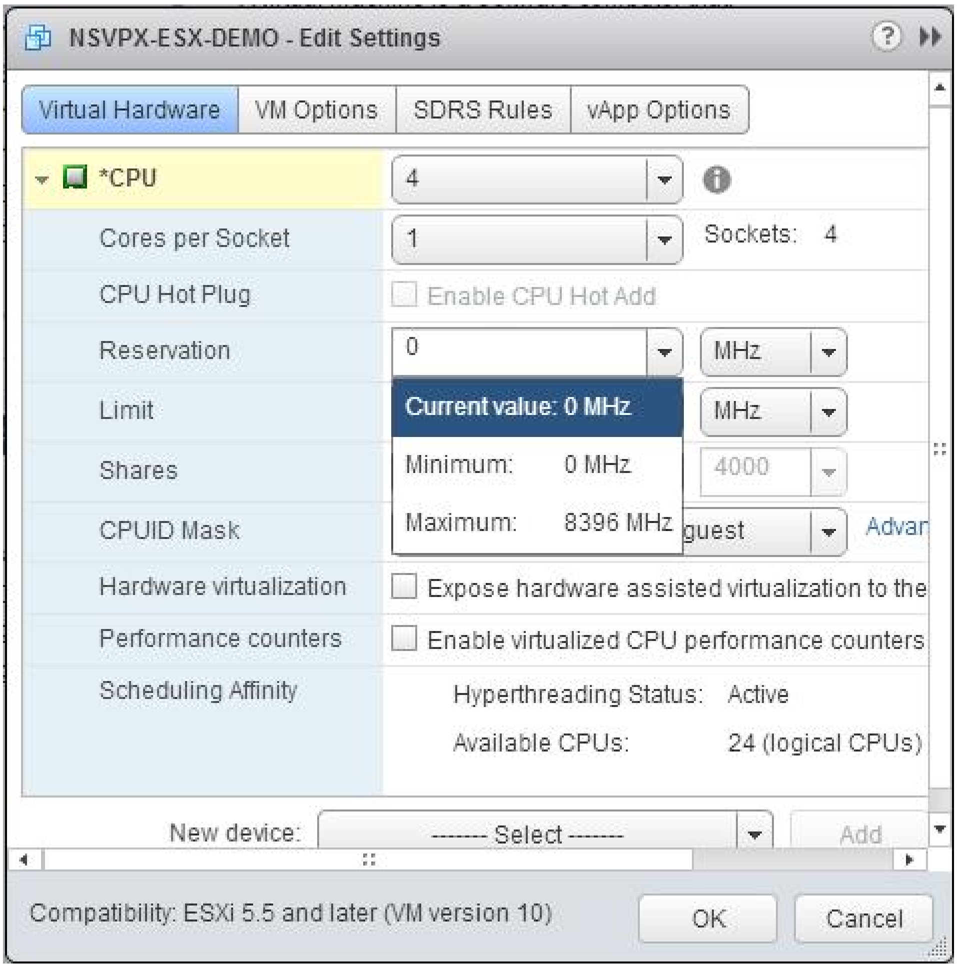
   a. In the CPU drop-down list, select the number of CPUs to assign to the virtual appliance.

b. In the Cores per Socket drop-down list, select the number of sockets.

c. (Optional) In the CPU Hot Plug field, select or unselect the Enable CPU Hot Add checkbox.

Note: Citrix recommends accepting the default (disabled).

d. In the Reservation drop-down list, select the number that is shown as the maximum value.



*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation: 0 MHz

Current value: 0 MHz

Minimum: 0 MHz

Maximum: 8396 MHz

Limit: MHz

Shares: 4000

CPUID Mask: guest Advan

Hardware virtualization: ☐ Expose hardware assisted virtualization to the

Performance counters: ☐ Enable virtualized CPU performance counters

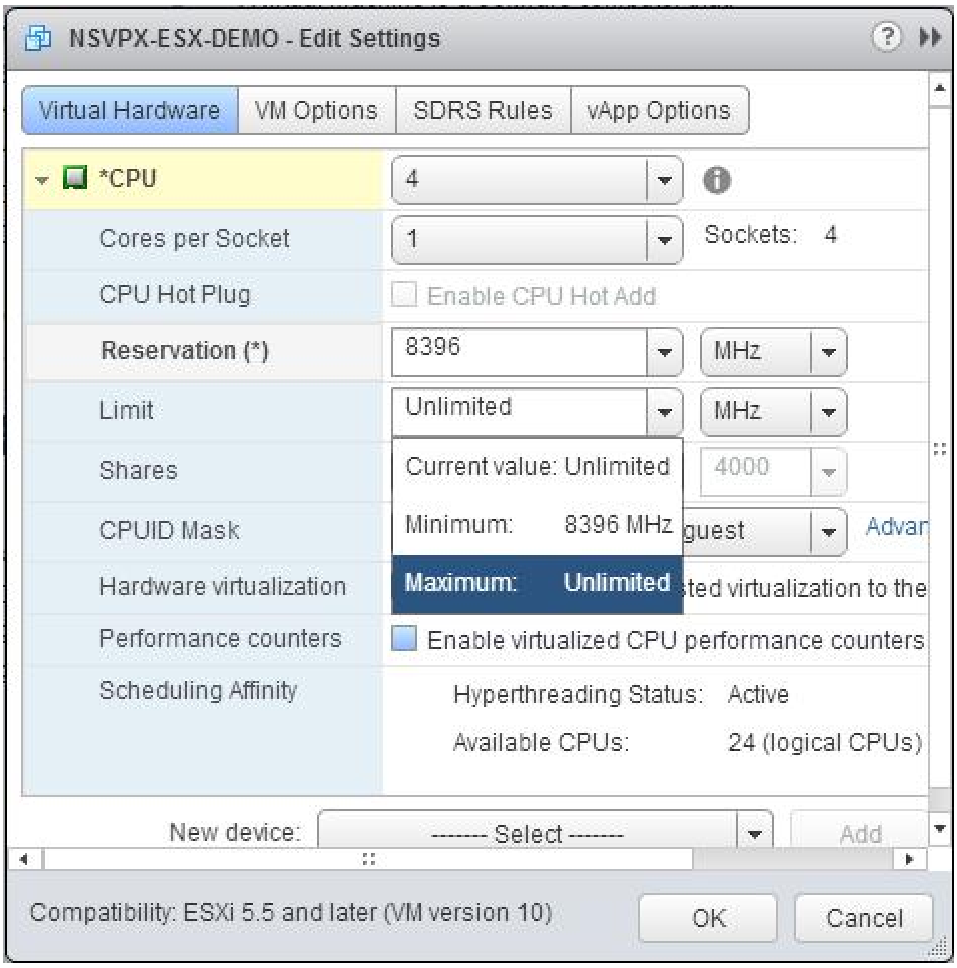
Scheduling Affinity: Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

New device: ------- Select -------

Compatibility: ESXi 5.5 and later (VM version 10)

e. In the Limit drop-down list, select the number that is shown as the maximum value.


Virtual Hardware
VM Options
SDRS Rules
vApp Options
New device:
------- Select -------
Add
OK
Cancel

NSVPX-ESX-DEMO - Edit Settings

*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation (*): 8396 MHz

Limit: Unlimited MHz

Current value: Unlimited

Minimum: 8396 MHz

Maximum: Unlimited

Shares: 4000

CPUID Mask: guest Advan

Hardware virtualization: ted virtualization to the

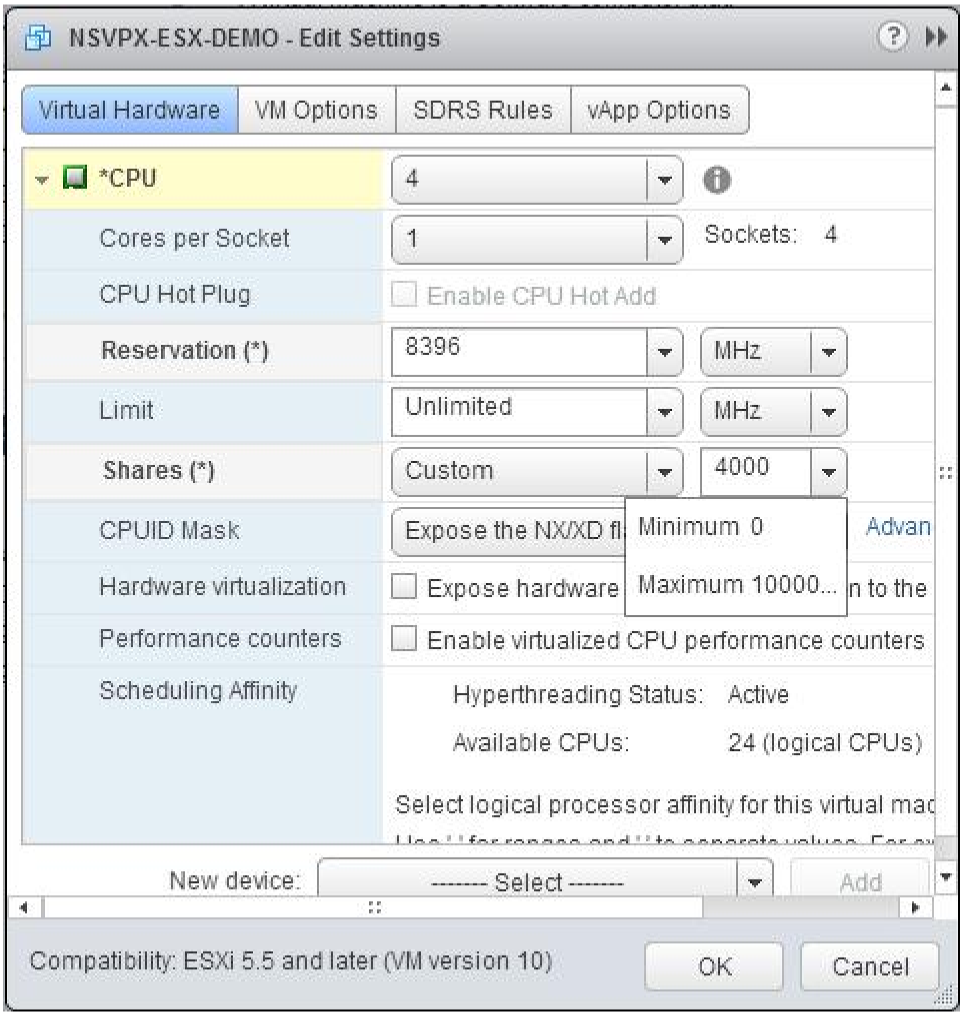
Performance counters: ☐ Enable virtualized CPU performance counters

Scheduling Affinity: Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

Compatibility: ESXi 5.5 and later (VM version 10)

f. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
New device:
------- Select -------
Add
OK
Cancel

*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation (*): 8396 MHz

Limit: Unlimited MHz

Shares (*): Custom 4000

Minimum 0

Maximum 10000...

CPUID Mask: Expose the NX/XD fl Advan

Hardware virtualization: ☐ Expose hardware n to the

Performance counters: ☐ Enable virtualized CPU performance counters

Scheduling Affinity: Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

Select logical processor affinity for this virtual mac

Compatibility: ESXi 5.5 and later (VM version 10)

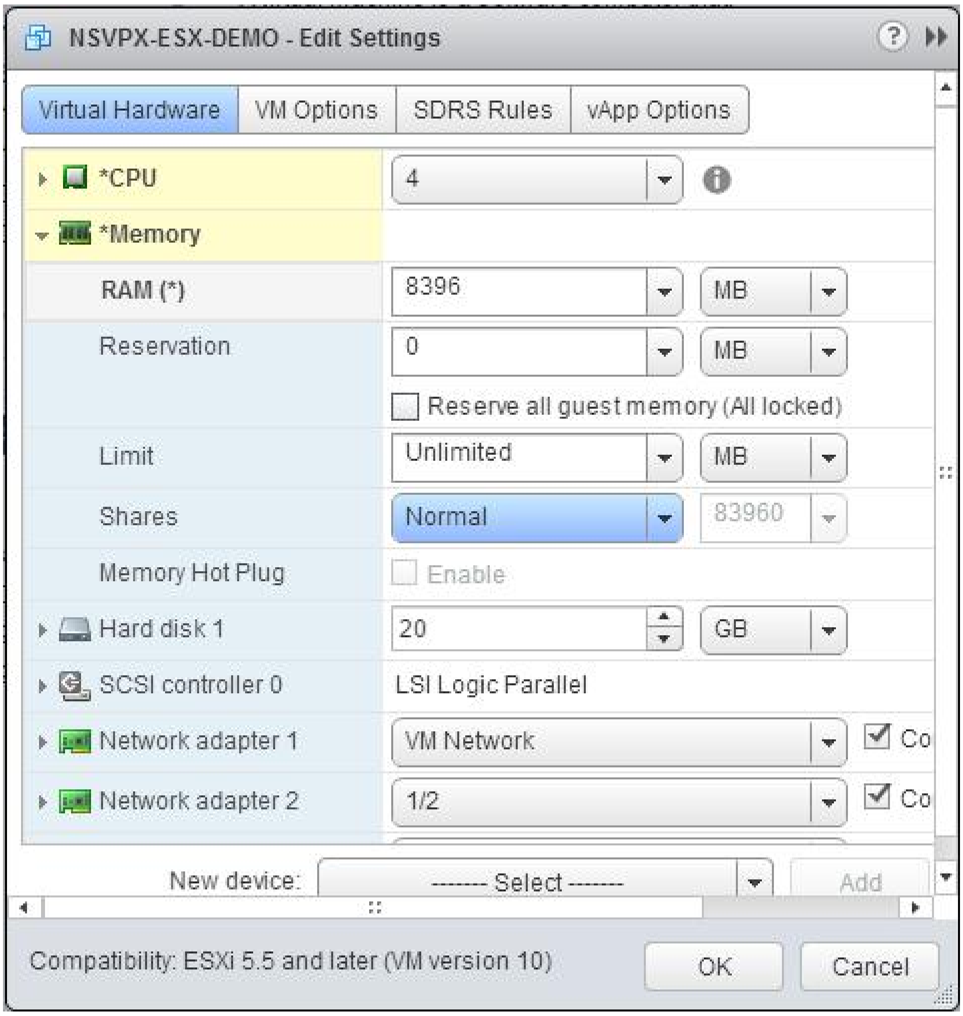
6. In the Memory section, update the following:

   - Size of RAM
   - Reservations
   - Limit
   - Shares

   Set the values as follows:

   a. In the RAM drop-down list, select the size of the RAM. It should be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the RAM should be 4 x 2 GB = 8 GB.

Note: For an Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



NSVPX-ESX-DEMO - Edit Settings

*CPU: 4

*Memory

RAM (*): 8396 MB

Reservation: 0 MB

☐ Reserve all guest memory (All locked)

Limit: Unlimited MB

Shares: Normal 83960

Memory Hot Plug: ☐ Enable

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Co

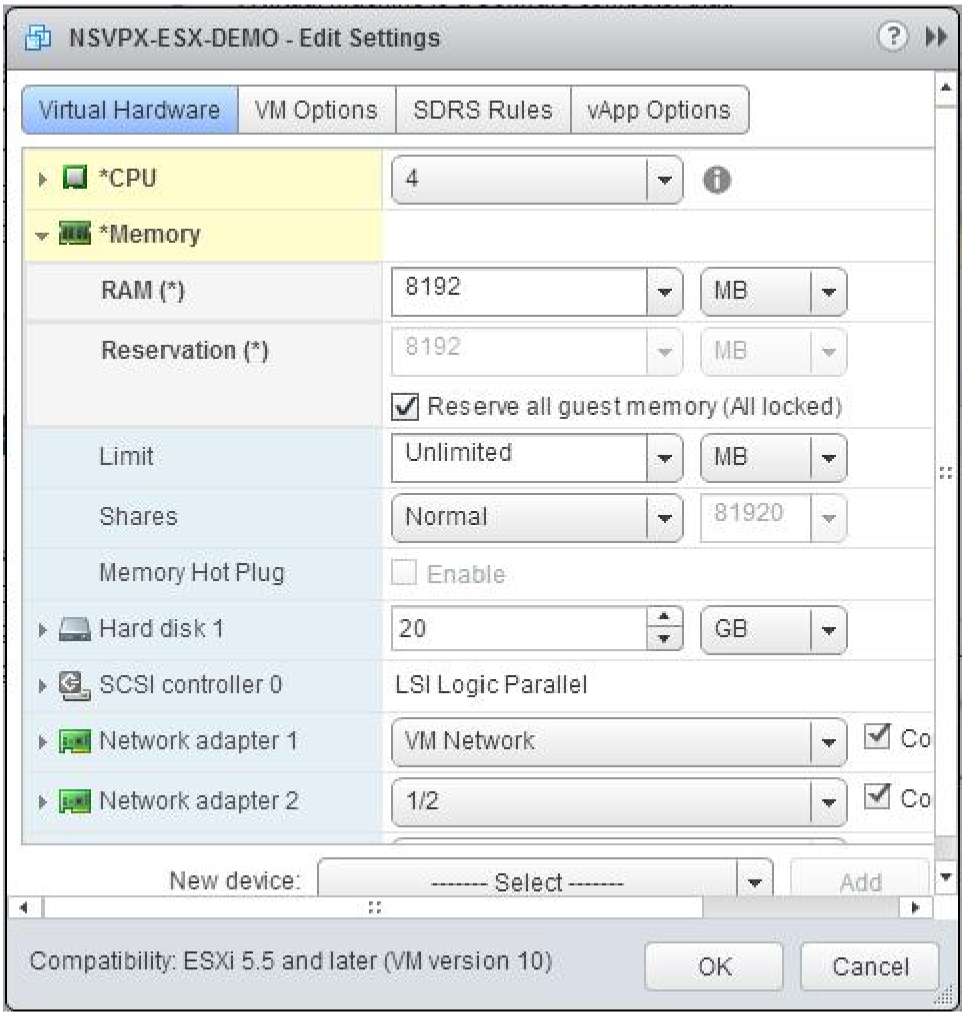
Network adapter 2: 1/2 ☑ Co

New device: ------- Select -------

Compatibility: ESXi 5.5 and later (VM version 10)

b. In the Reservation drop-down list, enter the value for the memory reservation, and select the Reserve all guest memory (All locked) checkbox. The memory reservation should be the number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation should be 4 x 2 GB = 8 GB.

Note: For an Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.

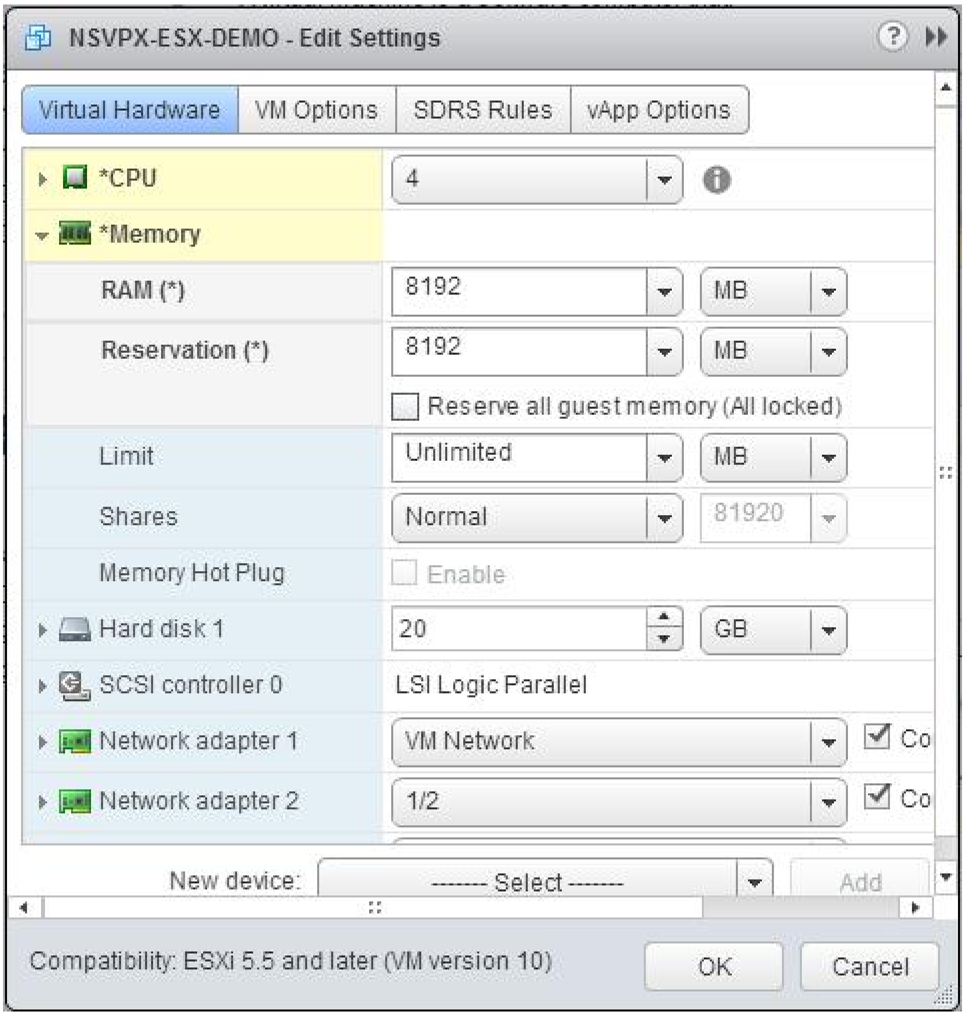

Virtual Hardware
VM Options
SDRS Rules
vApp Options
OK
Cancel

NSVPX-ESX-DEMO - Edit Settings

*CPU	4	
*Memory		
RAM (*)	8192	MB
Reservation (*)	8192	MB
	☑ Reserve all guest memory (All locked)	
Limit	Unlimited	MB
Shares	Normal	81920
Memory Hot Plug	☐ Enable	
Hard disk 1	20	GB
SCSI controller 0	LSI Logic Parallel	
Network adapter 1	VM Network	☑ Co
Network adapter 2	1/2	☑ Co
New device:	------- Select -------	Add

Compatibility: ESXi 5.5 and later (VM version 10)

c. In the Limit drop-down list, select the number that is shown as the maximum value.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

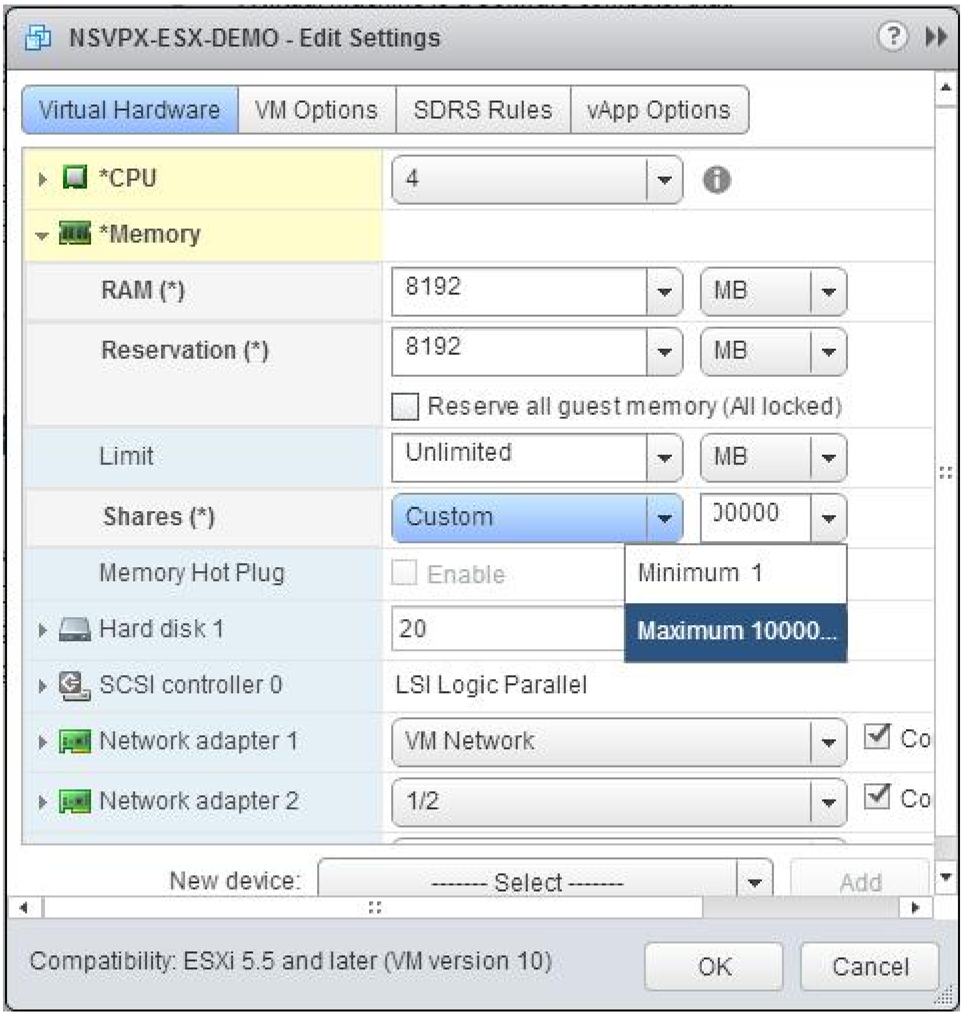
*CPU	4	
*Memory		
RAM (*)	8192	MB
Reservation (*)	8192	MB
	☐ Reserve all guest memory (All locked)	
Limit	Unlimited	MB
Shares	Normal	81920
Memory Hot Plug	☐ Enable	
Hard disk 1	20	GB
SCSI controller 0	LSI Logic Parallel	
Network adapter 1	VM Network	☑ Co
Network adapter 2	1/2	☑ Co

New device:
------- Select -------
Add

Compatibility: ESXi 5.5 and later (VM version 10)

OK
Cancel

d. In the Shares drop-down lists, select Custom and the number that is shown as the maximum value.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
Minimum 1
Maximum 10000...
OK
Cancel

*CPU: 4

*Memory

RAM (*): 8192 MB

Reservation (*): 8192 MB

☐ Reserve all guest memory (All locked)

Limit: Unlimited MB

Shares (*): Custom)0000

Memory Hot Plug: ☐ Enable

Hard disk 1: 20

SCSI controller 0: LSI Logic Parallel

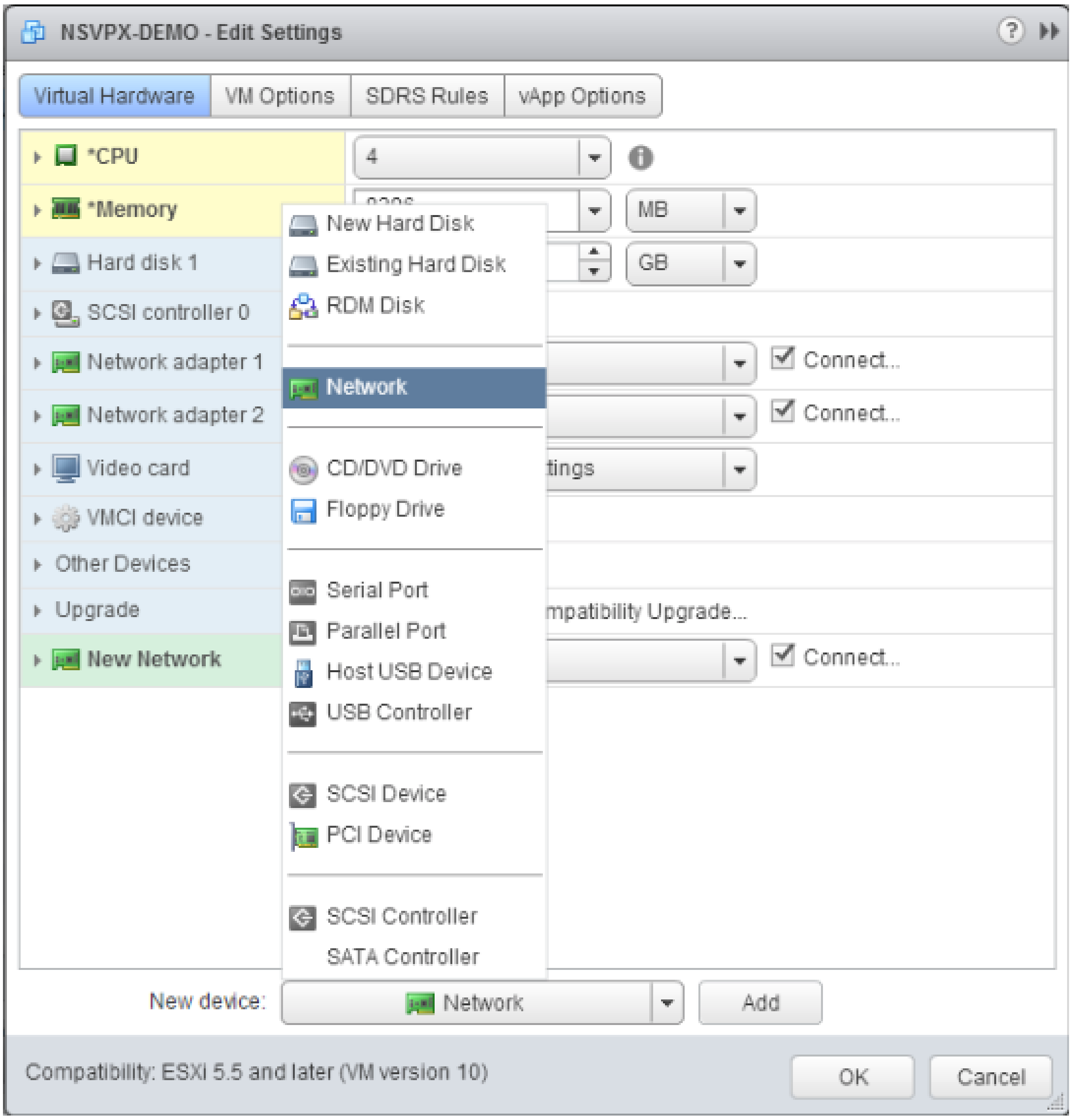
Network adapter 1: VM Network ☑ Co

Network adapter 2: 1/2 ☑ Co

New device: ------- Select ------- Add

Compatibility: ESXi 5.5 and later (VM version 10)

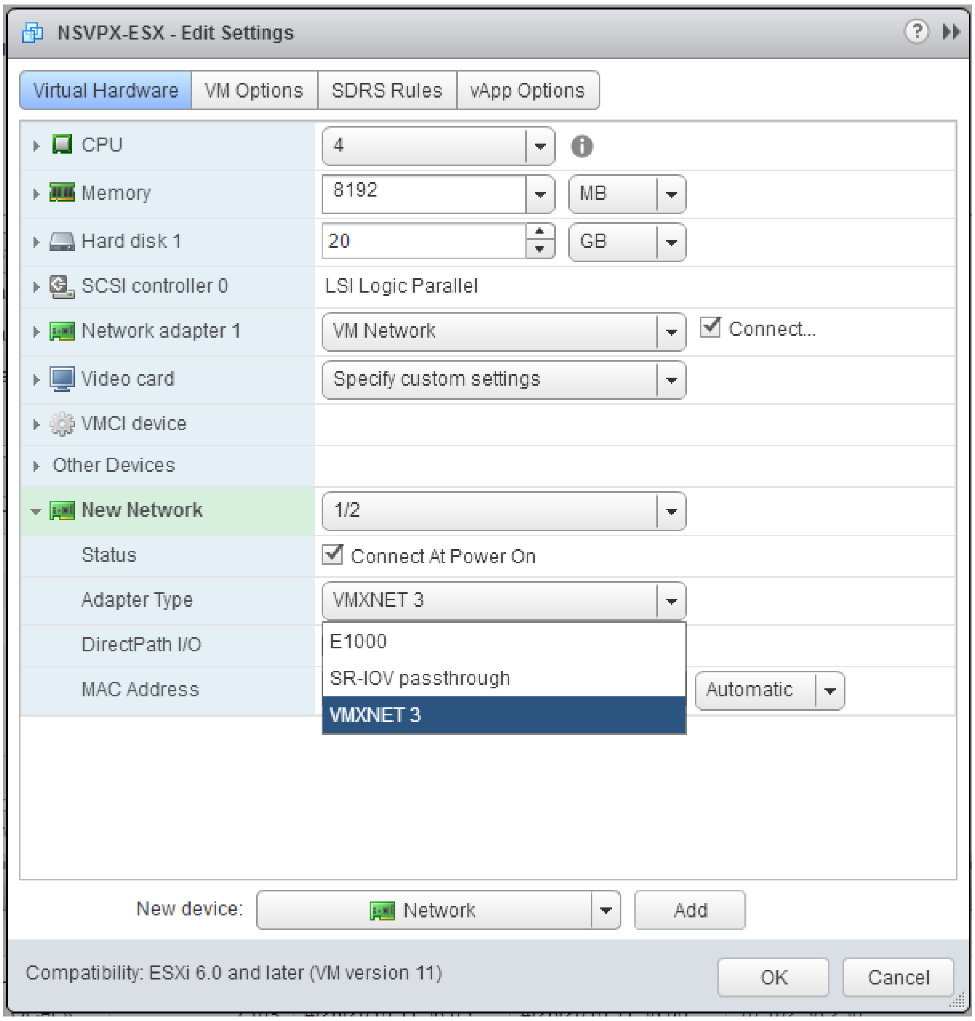
7. Add a VMXNET3 network interface. From the New device drop-down list, select Network and click Add.


NSVPX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
*CPU
4
*Memory
MB
Hard disk 1
GB
SCSI controller 0
Network adapter 1
Connect...
Network adapter 2
Connect...
Video card
VMCI device
Other Devices
Upgrade
New Network
Connect...
New Hard Disk
Existing Hard Disk
RDM Disk
Network
CD/DVD Drive
Floppy Drive
Serial Port
Parallel Port
Host USB Device
USB Controller
SCSI Device
PCI Device
SCSI Controller
SATA Controller
New device:
Network
Add
Compatibility: ESXi 5.5 and later (VM version 10)
OK
Cancel

8. In the New Network section, from the drop-down list, select the network interface, and do the following:

a. In the Adapter Type drop-down list, select VMXNET3.

> **Important**
>
> The default E1000 network interface and VMXNET3 cannot coexist, make sure that you remove the E1000 network interface and use VMXNET3 (0/1) as the management interface.



NSVPX-ESX - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

CPU: 4

Memory: 8192 MB

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

New Network: 1/2

Status: ☑ Connect At Power On

Adapter Type: VMXNET 3

DirectPath I/O

MAC Address: Automatic

E1000
SR-IOV passthrough
VMXNET 3
New device:
Network
Add

Compatibility: ESXi 6.0 and later (VM version 11)

OK
Cancel

9. Click OK.

10. Power on the NetScaler VPX instance.

11. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

```
> show interface summary
```

The output should show all the interfaces that you configured:

```
> show interface summary
----------------------------------------------------------------------

      Interface  MTU        MAC                    Suffix
----------------------------------------------------------------------

1      0/1        1500       00:0c:29:89:1d:0e     NetScaler Vir...rface,
     VMXNET3
```

```
2     1/1        9000       00:0c:29:89:1d:18    NetScaler Vir...rface,
    VMXNET3
3     1/2        9000       00:0c:29:89:1d:22    NetScaler Vir...rface,
    VMXNET3
4     LO/1       9000       00:0c:29:89:1d:0e    Netscaler Loopback
    interface
```

> **Note**
>
> After you add a VMXNET3 interface and restart the NetScaler VPX appliance, the VMWare ESX hypervisor might change the order in which the NIC is presented to the VPX appliance. So, network adapter 1 might not always remain 0/1, resulting in loss of management connectivity to the VPX appliance. To avoid this issue, change the virtual network of the network adapter accordingly.
>
> This is a VMWare ESX hypervisor limitation.

Configure a NetScaler VPX instance to use SR-IOV network interface

September 21, 2020

After you have installed and configured the NetScaler VPX instance on VMware ESX, you can use the VMware vSphere web client to configure the virtual appliance to use single root I/O v irtualization (SR-IOV) network interfaces.

Limitations

A NetScaler VPX configured with SR-IOV network interface has the following limitations:

- The following features are not supported on SR-IOV interfaces using Intel 82599 10G NIC on ESX VPX:
  - L2 mode switching
  - Static Link Aggregation and LACP
  - Clustering
  - Admin partitioning [Shared VLAN mode]
  - High Availability [Active - Active mode]
  - Jumbo frames
  - IPv6
- The following features are not supported for on SR-IOV interface with an Intel 82599 10G NIC on KVM VPX:
  - Static Link Aggregation and LACP
  - L2 mode switching

- Clustering
- Admin partitioning [Shared VLAN mode]
- High Availability [Active – Active mode]
- Jumbo frames
- IPv6
- VLAN configuration on Hypervisor for SR-IOV VF interface through "ip link" command is not supported

Prerequisite

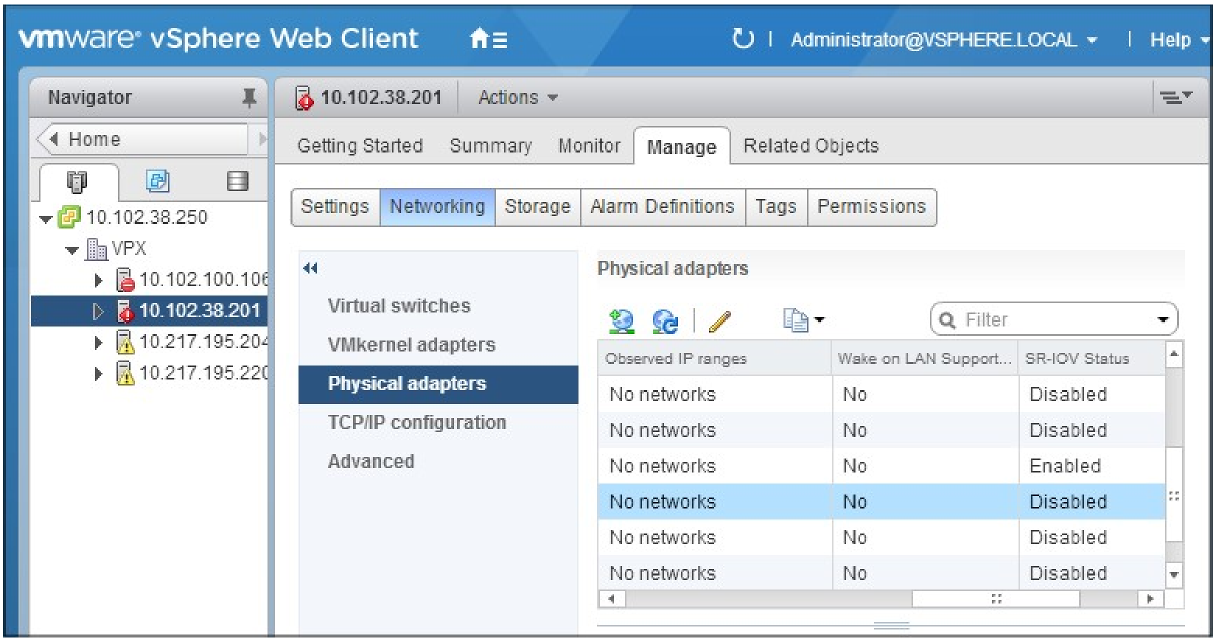
Make sure that you:

- Add the Intel 82599 Network Interface Card (NIC) to the ESX Host. IXGBE driver version 3.7.13.7.14iov is recommended.

- Enable SR-IOV on the host physical adapter, as follows:

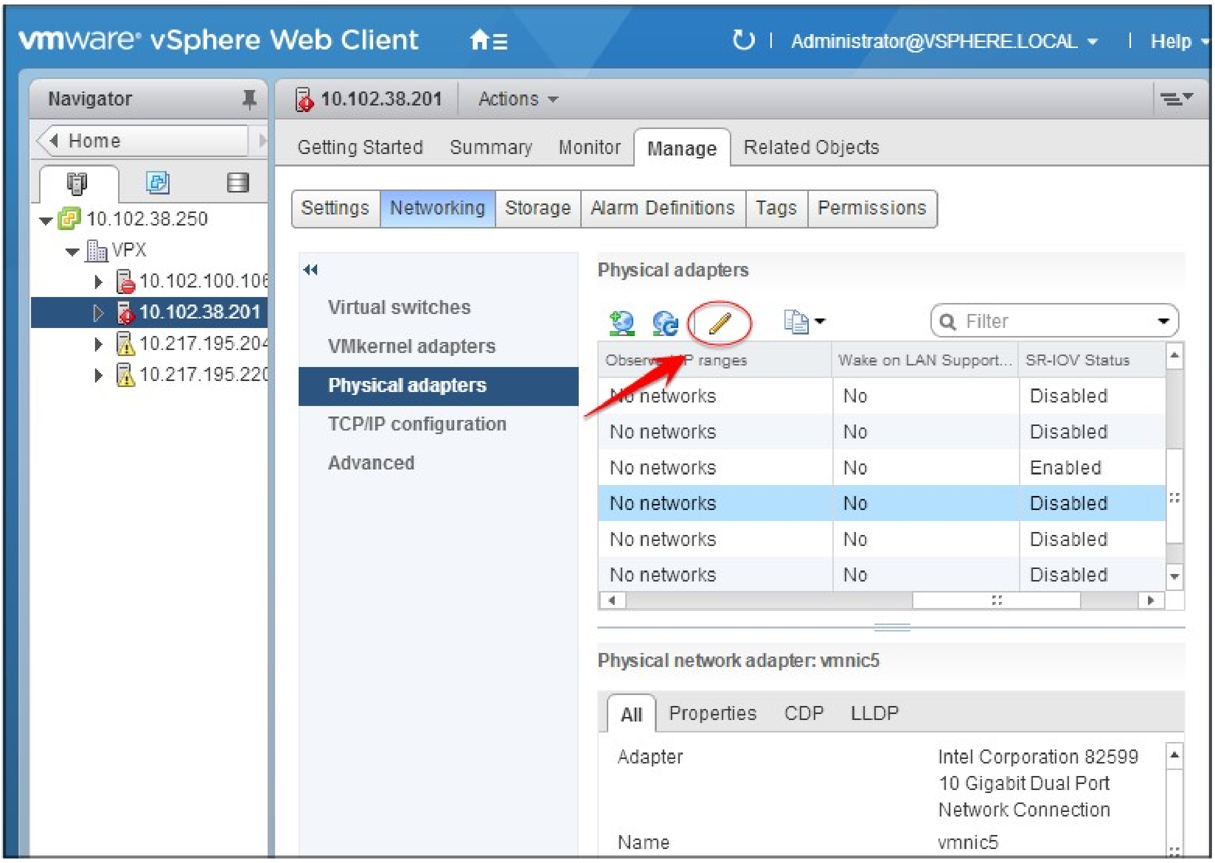
1. In the vSphere Web Client, navigate to the Host.

2. On the **Manage > Networking** tab, select **Physical adapters**. The SR-IOV Status field shows whether a physical adapter supports SR-IOV.


vmware® vSphere Web Client
Administrator@VSPHERE.LOCAL
Help
Navigator
Home
10.102.38.250
VPX
10.102.100.106
10.102.38.201
10.217.195.204
10.217.195.220
10.102.38.201
Actions
Getting Started
Summary
Monitor
Manage
Related Objects
Settings
Networking
Storage
Alarm Definitions
Tags
Permissions
Virtual switches
VMkernel adapters
Physical adapters
TCP/IP configuration
Advanced
Physical adapters
Filter

Observed IP ranges	Wake on LAN Support...	SR-IOV Status
No networks	No	Disabled
No networks	No	Disabled
No networks	No	Enabled
No networks	No	Disabled
No networks	No	Disabled
No networks	No	Disabled

3. Select the physical adapter, and then click the pencil icon to open the **Edit Settings** dialog box.


vmware vSphere Web Client
Administrator@VSPHERE.LOCAL
Help
Navigator
Home
10.102.38.250
VPX
10.102.100.106
10.102.38.201
10.217.195.204
10.217.195.220
10.102.38.201
Actions
Getting Started
Summary
Monitor
Manage
Related Objects
Settings
Networking
Storage
Alarm Definitions
Tags
Permissions
Virtual switches
VMkernel adapters
Physical adapters
TCP/IP configuration
Advanced
Physical adapters
Filter

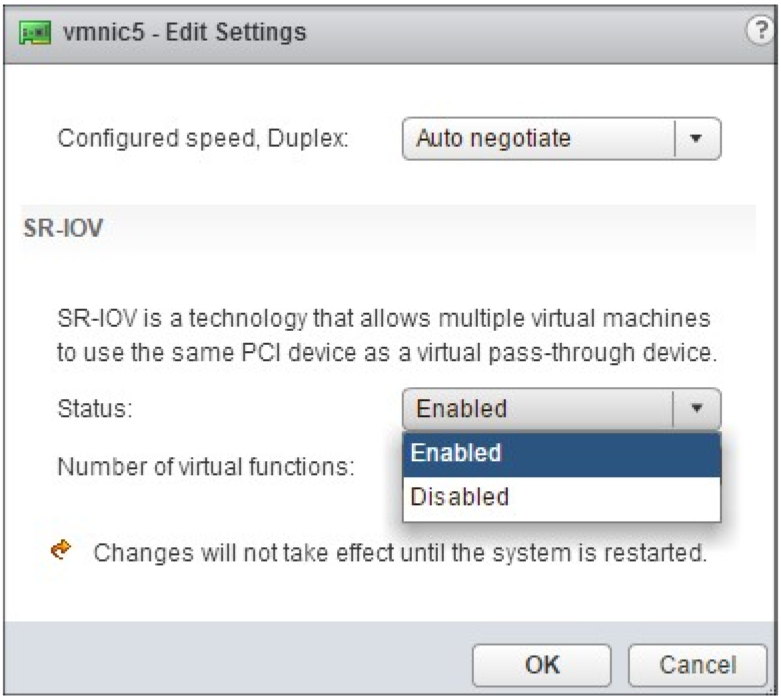
Observed IP ranges	Wake on LAN Support...	SR-IOV Status
No networks	No	Disabled
No networks	No	Disabled
No networks	No	Enabled
No networks	No	Disabled
No networks	No	Disabled
No networks	No	Disabled

Physical network adapter: vmnic5

All
Properties
CDP
LLDP

Adapter	Intel Corporation 82599 10 Gigabit Dual Port Network Connection
Name	vmnic5

4. Under SR-IOV, select **Enabled** from the **Status** drop-down list.


vmnic5 - Edit Settings
SR-IOV
Enabled
Disabled
OK
Cancel

Configured speed, Duplex: Auto negotiate

SR-IOV is a technology that allows multiple virtual machines to use the same PCI device as a virtual pass-through device.

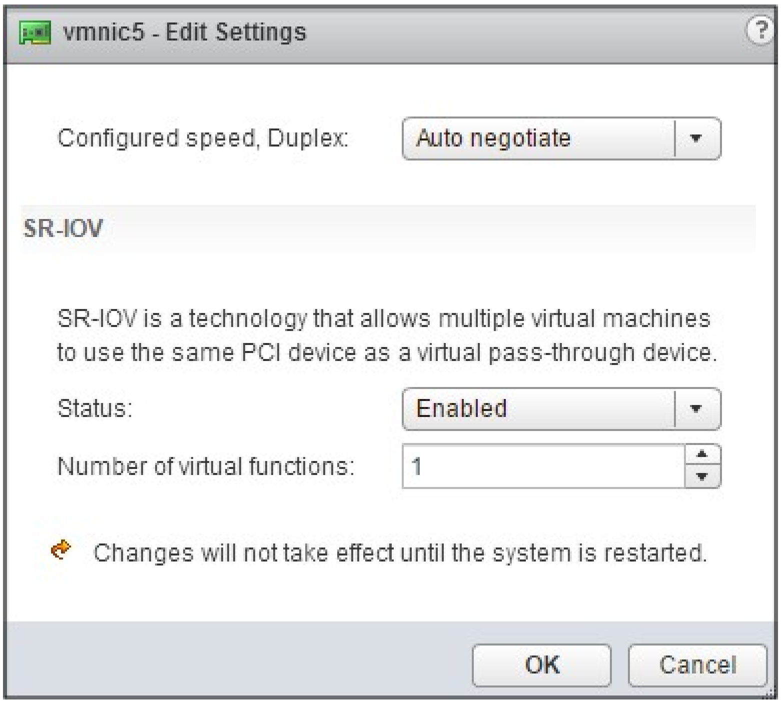
Status: Enabled

Number of virtual functions:

Changes will not take effect until the system is restarted.

5. In the **Number of virtual functions** field, enter the number of virtual functions that you want to

configure for the adapter.



Configured speed, Duplex: Auto negotiate

SR-IOV

SR-IOV is a technology that allows multiple virtual machines to use the same PCI device as a virtual pass-through device.

Status: Enabled

Number of virtual functions: 1

Changes will not take effect until the system is restarted.

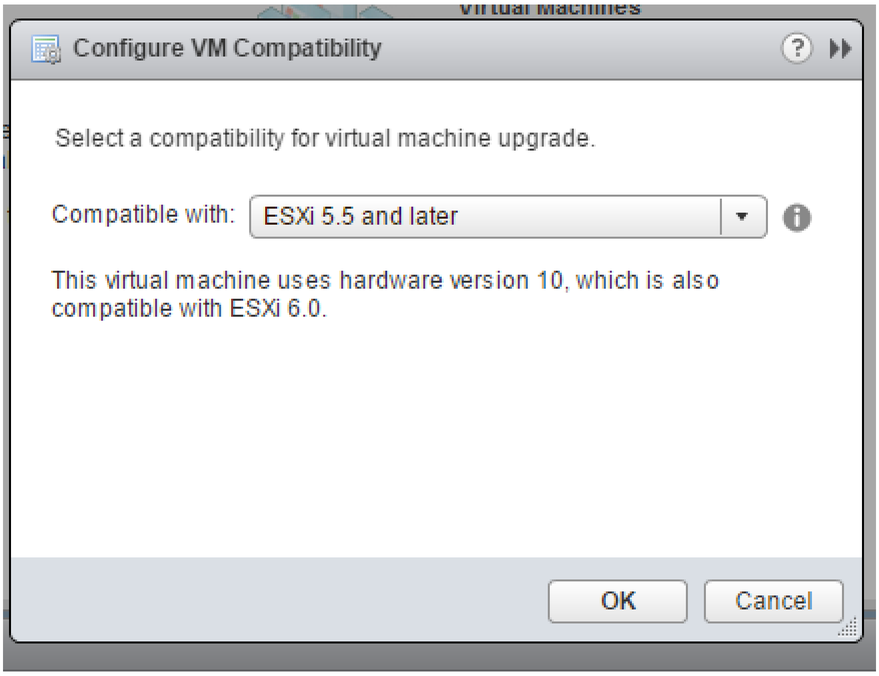
6. Click **OK**.
7. Restart the host.

- Create a Distributed Virtual Switch (DVS) and Portgroups. For instructions, see the VMware Documentation.

> **Note**
>
> Citrix has qualified the SR-IOV configuration on DVS and Portgroups only.

To configure NetScaler VPX instances to use SR-IOV network interface by using VMware vSphere Web Client:

1. In the vSphere Web Client, select **Hosts and Clusters**.
2. Upgrade the Compatibility setting of the NetScaler VPX instance to ESX 5.5 or later, as follows:
   a. Power off the NetScaler VPX instance.
   b. Right-click the NetScaler VPX instance and select **Compatibility > Upgrade VM Compatibility**.
   c. In the **Configure VM Compatibility** dialog box, select **ESXi 5.5 and later** from the **Compatible with** drop-down list and click **OK**.

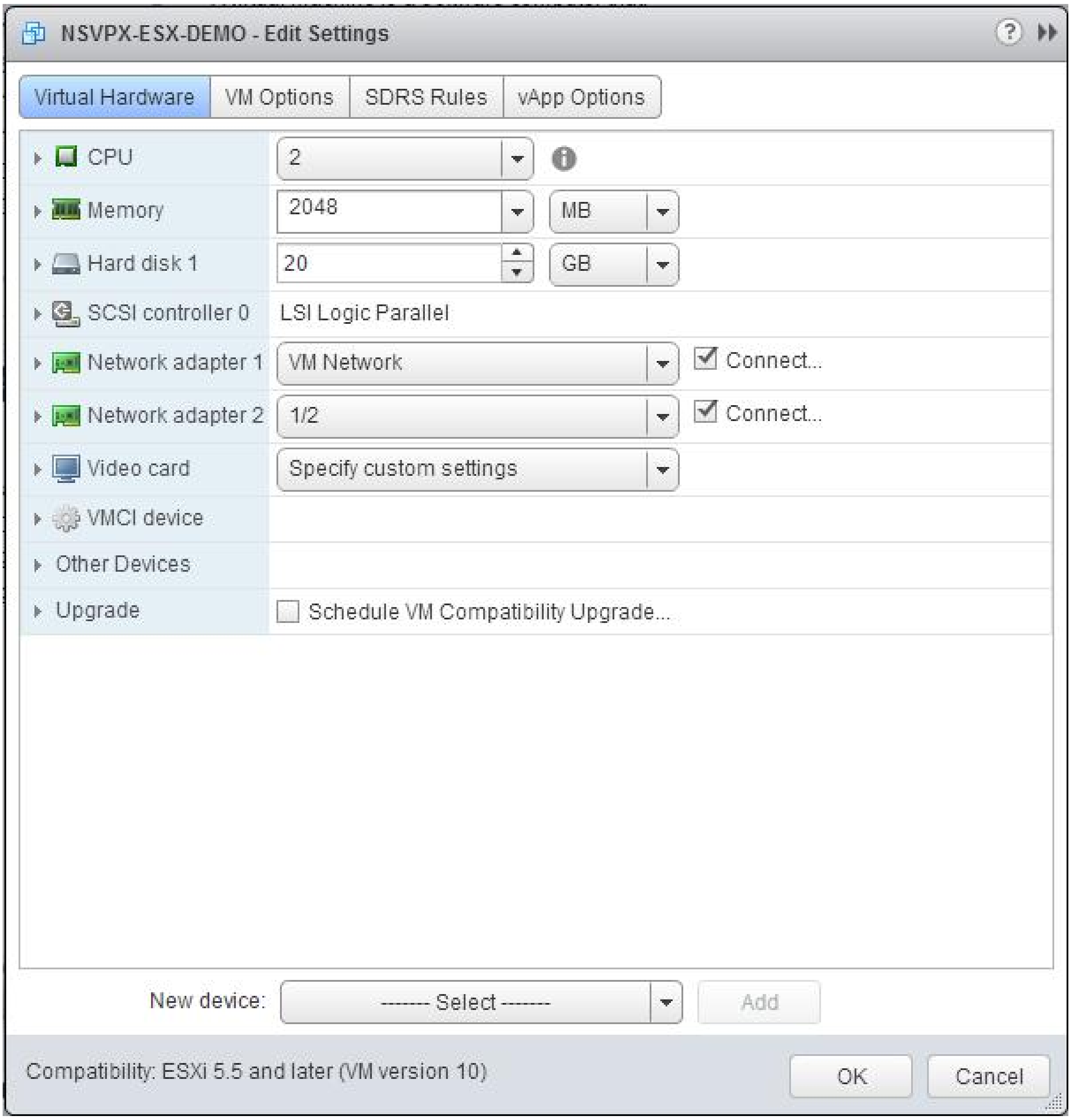

Configure VM Compatibility
OK
Cancel

Select a compatibility for virtual machine upgrade.

Compatible with: ESXi 5.5 and later

This virtual machine uses hardware version 10, which is also compatible with ESXi 6.0.

3. Right-click on the NetScaler VPX instance and click **Edit Settings**.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

CPU: 2

Memory: 2048 MB

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Network adapter 2: 1/2 ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

Upgrade: ☐ Schedule VM Compatibility Upgrade...

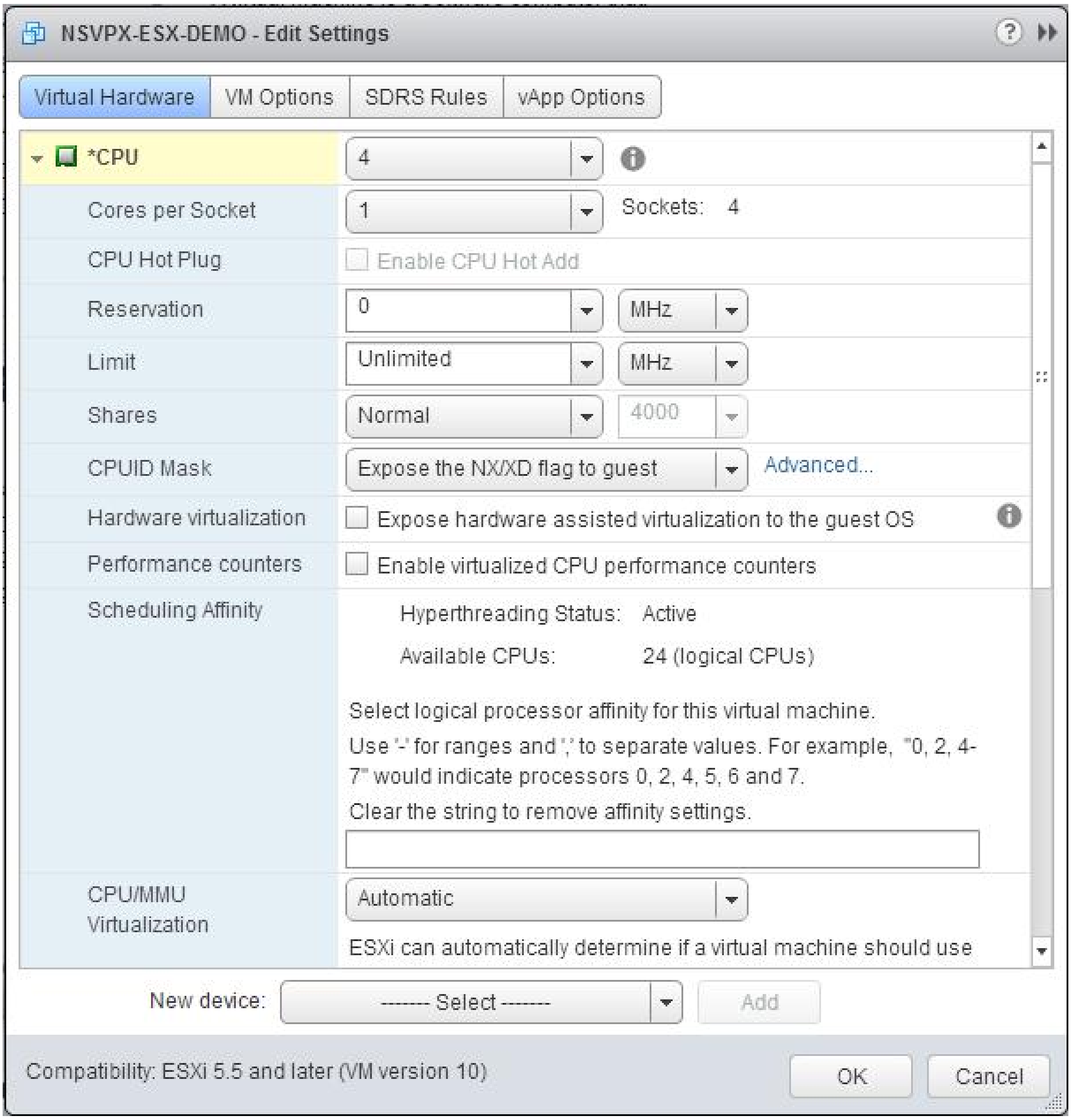
New device: ------- Select -------

Add

Compatibility: ESXi 5.5 and later (VM version 10)

OK
Cancel

4. In the **<virtual_appliance> - Edit Settings** dialog box, click the **CPU** section.



*CPU	4
Cores per Socket	1 Sockets: 4
CPU Hot Plug	☐ Enable CPU Hot Add
Reservation	0 MHz
Limit	Unlimited MHz
Shares	Normal 4000
CPUID Mask	Expose the NX/XD flag to guest Advanced...
Hardware virtualization	☐ Expose hardware assisted virtualization to the guest OS
Performance counters	☐ Enable virtualized CPU performance counters
Scheduling Affinity	Hyperthreading Status: Active Available CPUs: 24 (logical CPUs) Select logical processor affinity for this virtual machine. Use '-' for ranges and ',' to separate values. For example, "0, 2, 4-7" would indicate processors 0, 2, 4, 5, 6 and 7. Clear the string to remove affinity settings.
CPU/MMU Virtualization	Automatic ESXi can automatically determine if a virtual machine should use

Compatibility: ESXi 5.5 and later (VM version 10)

5. In the **CPU** section, update the following settings:

   - Number of CPUs

   - Number of Sockets

   - Reservations

   - Limit

   - Shares

   Set the values as follows:

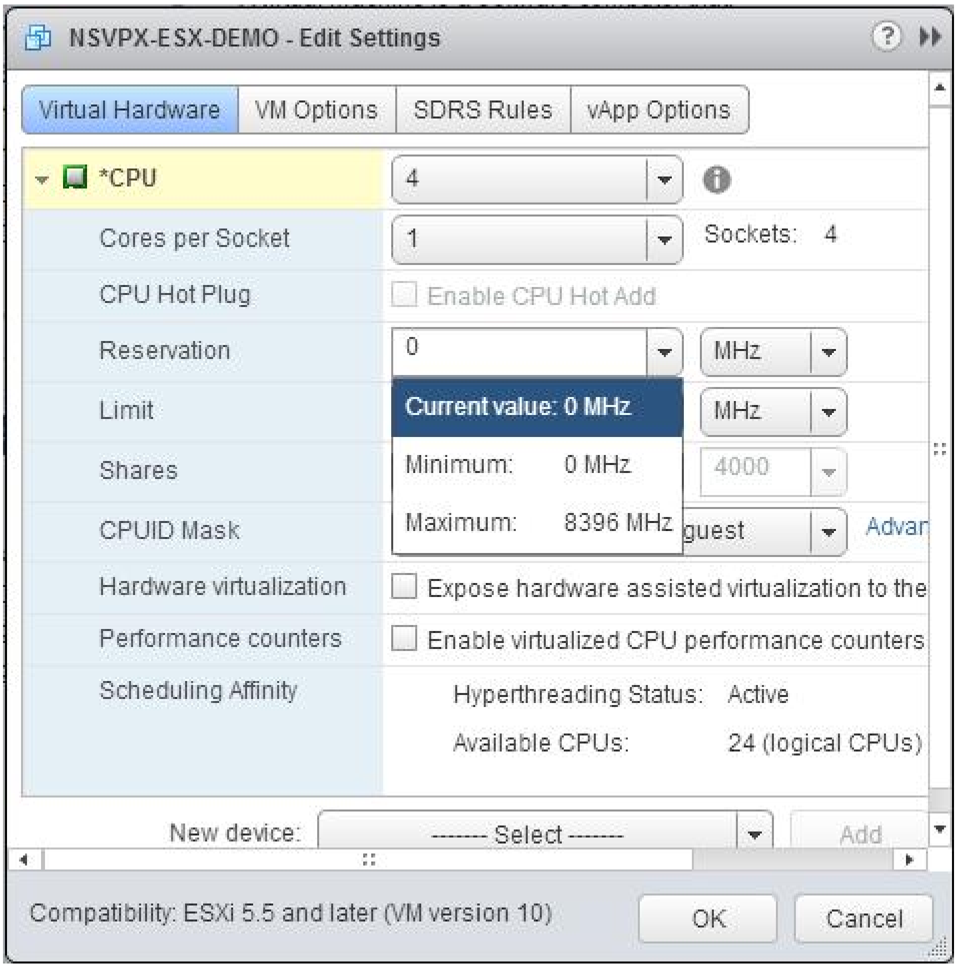
   a. In the **CPU** drop-down list, select the number of CPUs to assign to the virtual appliance.

b. In the **Cores per Socket** drop-down list, select the number of sockets.

c. (Optional) In the **CPU Hot Plug** field, select or clear the **Enable CPU Hot Add** check box.

Note: Citrix recommends accepting the default (disabled).

d. In the **Reservation** drop-down list, select the number that is shown as the maximum value.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
New device:
------- Select -------
Add
OK
Cancel

*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation: 0 MHz

Current value: 0 MHz

Minimum: 0 MHz

Maximum: 8396 MHz

Limit: MHz

Shares: 4000

CPUID Mask: guest Advan

Hardware virtualization: ☐ Expose hardware assisted virtualization to the

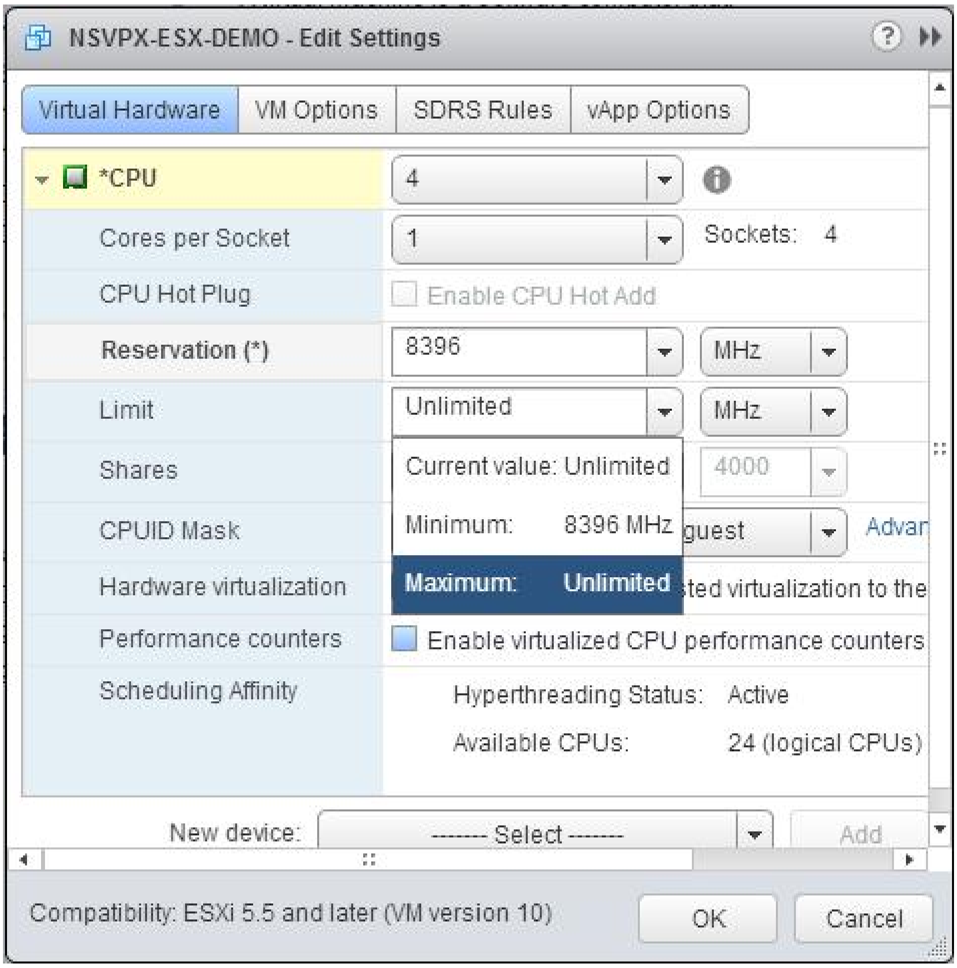
Performance counters: ☐ Enable virtualized CPU performance counters

Scheduling Affinity: Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

Compatibility: ESXi 5.5 and later (VM version 10)

e. In the **Limit** drop-down list, select the number that is shown as the maximum value.



NSVPX-ESX-DEMO - Edit Settings

*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation (*): 8396 MHz

Limit: Unlimited MHz

Current value: Unlimited

Minimum: 8396 MHz

Maximum: Unlimited

Shares: 4000

CPUID Mask: guest

Hardware virtualization: ted virtualization to the

Performance counters: ☐ Enable virtualized CPU performance counters

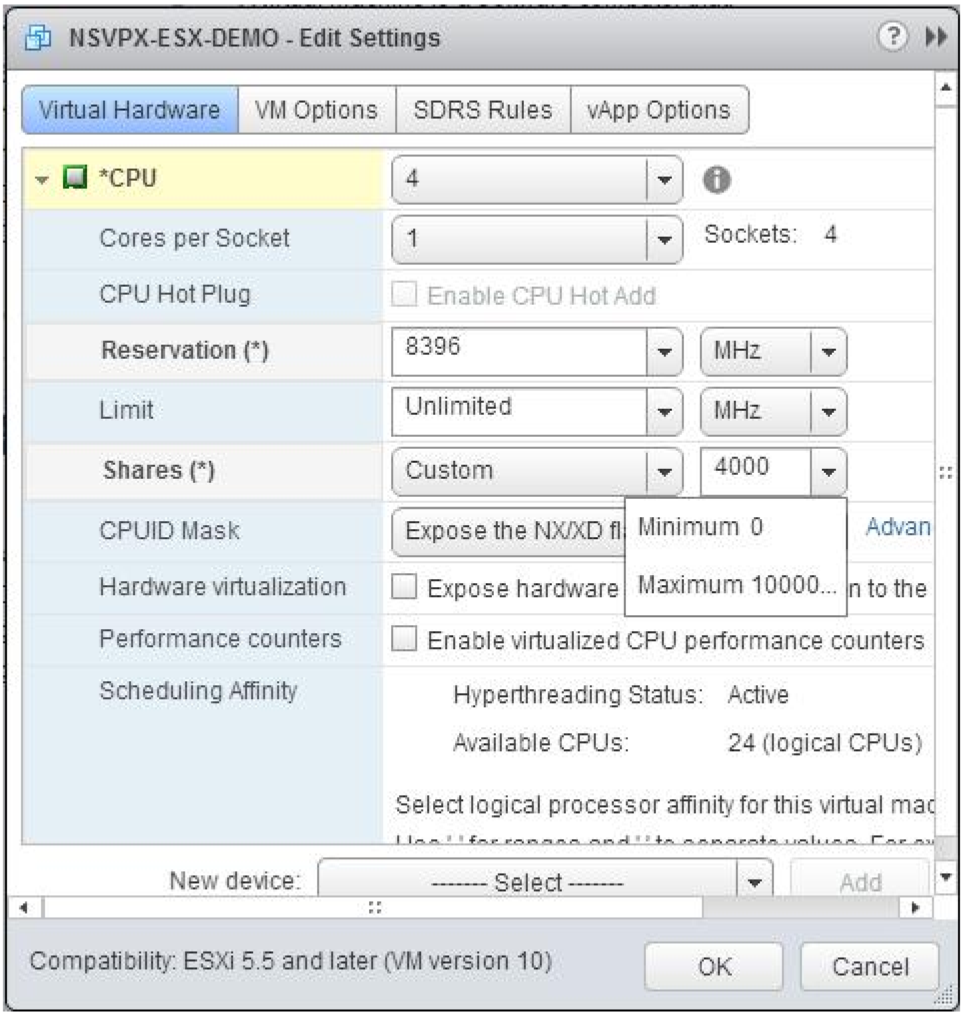
Scheduling Affinity: Hyperthreading Status: Active

Available CPUs: 24 (logical CPUs)

New device: ------- Select ------- Add

Compatibility: ESXi 5.5 and later (VM version 10)

f. In the **Shares** drop-down lists, select **Custom** and the number that is shown as the maximum value.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
New device:
------- Select -------
Add
OK
Cancel

*CPU: 4

Cores per Socket: 1 Sockets: 4

CPU Hot Plug: ☐ Enable CPU Hot Add

Reservation (*): 8396 MHz

Limit: Unlimited MHz

Shares (*): Custom 4000 (Minimum 0, Maximum 10000...)

CPUID Mask: Expose the NX/XD fl

Hardware virtualization: ☐ Expose hardware

Performance counters: ☐ Enable virtualized CPU performance counters

Scheduling Affinity: Hyperthreading Status: Active; Available CPUs: 24 (logical CPUs)

Select logical processor affinity for this virtual mac

Compatibility: ESXi 5.5 and later (VM version 10)

6. In the **Memory** section, update the following settings:

   - Size of RAM

   - Reservations

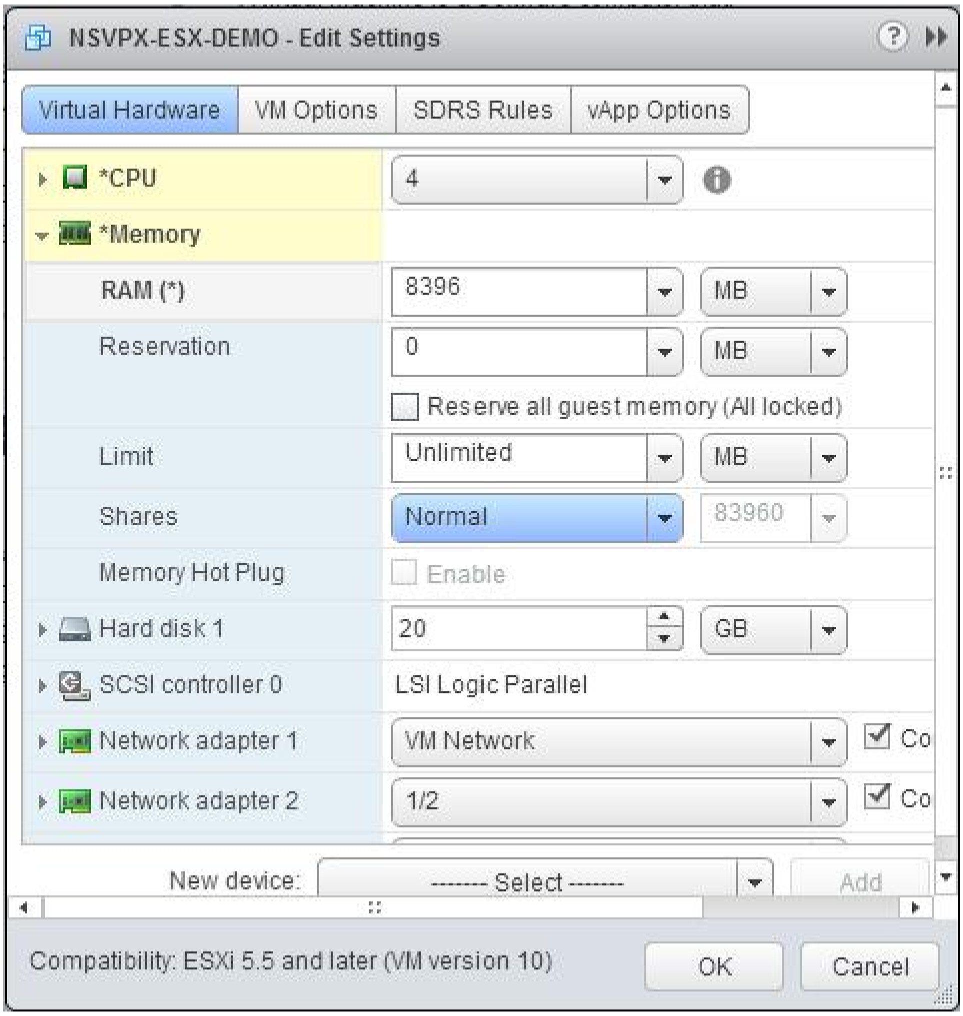
   - Limit

   - Shares

   Set the values as follows:

   a. In the **RAM** drop-down list, select the size of the RAM. It should be number of vCPUs x 2 GB. For example, if the number of vCPU is 4 then RAM = 4 x 2 GB = 8 GB.

Note: For Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.



NSVPX-ESX-DEMO - Edit Settings

*CPU 4

*Memory

RAM (*) 8396 MB

Reservation 0 MB

☐ Reserve all guest memory (All locked)

Limit Unlimited MB

Shares Normal 83960

Memory Hot Plug ☐ Enable

Hard disk 1 20 GB

SCSI controller 0 LSI Logic Parallel

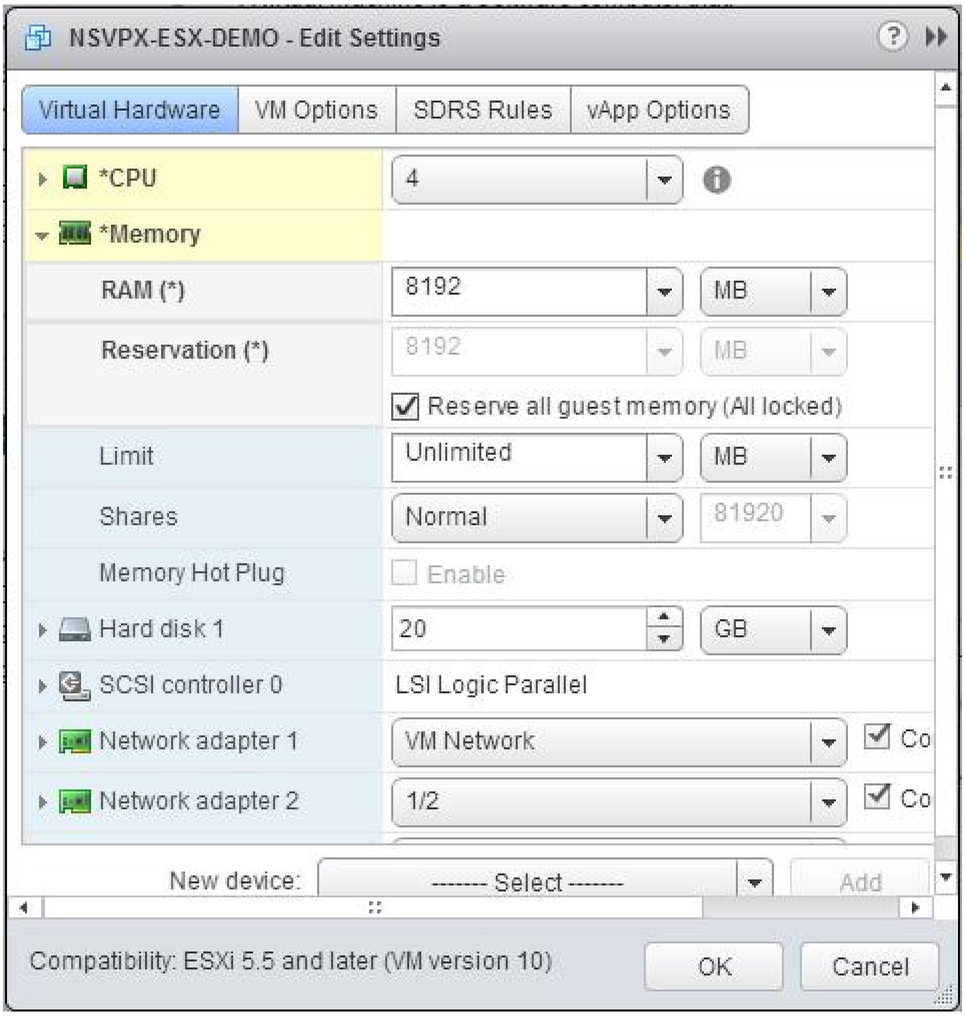
Network adapter 1 VM Network ☑ Co

Network adapter 2 1/2 ☑ Co

Compatibility: ESXi 5.5 and later (VM version 10)

b. In the **Reservation** drop-down list, enter the value for the memory reservation, and select the **Reserve all guest memory (All locked)** check box. The memory reservation should be number of vCPUs x 2 GB. For example, if the number of vCPUs is 4, the memory reservation should be 4 x 2 GB = 8 GB.

Note: For Enterprise or Platinum edition of the NetScaler VPX appliance, make sure that you allocate 4 GB of RAM to each vCPU. For example, if the number of vCPU is 4 then RAM = 4 x 4 GB = 16 GB.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
New device:
------- Select -------
Add
OK
Cancel

*CPU 4

*Memory

RAM (*) 8192 MB

Reservation (*) 8192 MB

☑ Reserve all guest memory (All locked)

Limit Unlimited MB

Shares Normal 81920

Memory Hot Plug ☐ Enable

Hard disk 1 20 GB

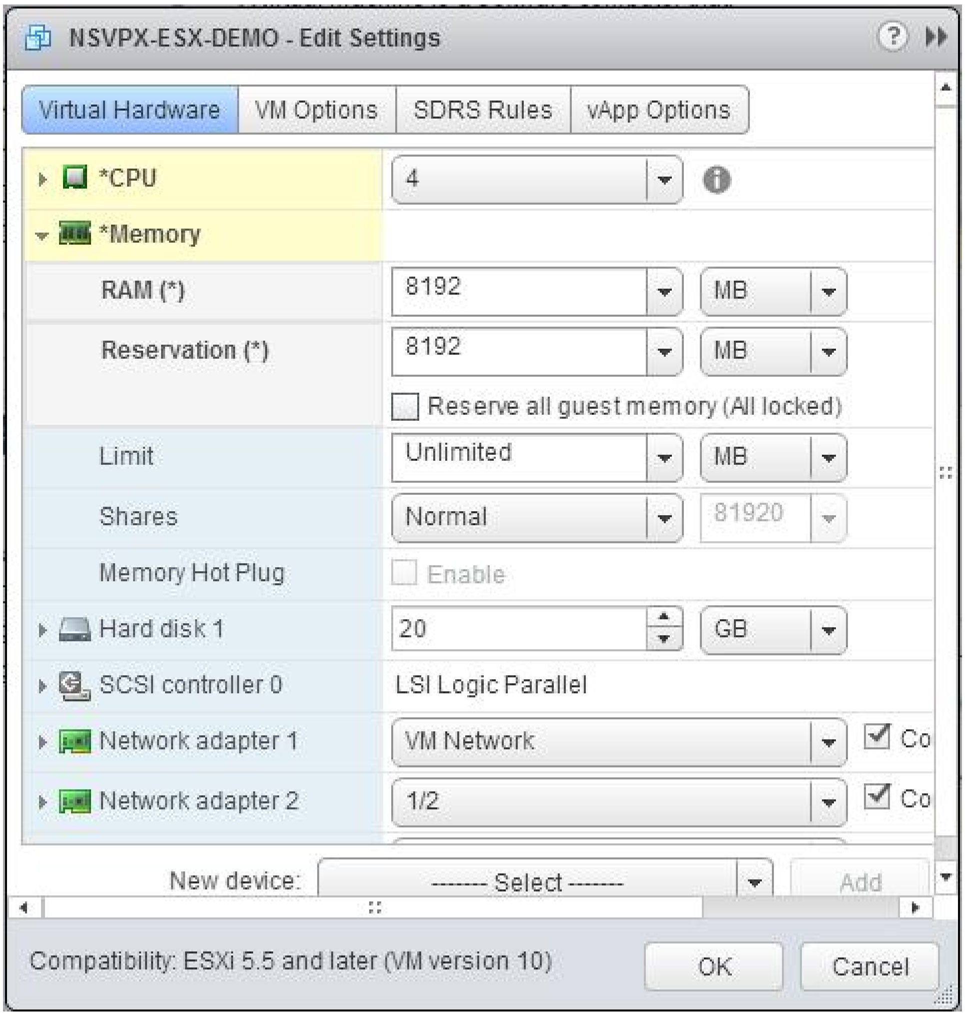
SCSI controller 0 LSI Logic Parallel

Network adapter 1 VM Network ☑ Co

Network adapter 2 1/2 ☑ Co

Compatibility: ESXi 5.5 and later (VM version 10)

c. In the **Limit** drop-down list, select the number that is shown as the maximum value.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware | VM Options | SDRS Rules | vApp Options

*CPU: 4

*Memory

RAM (*): 8192 MB

Reservation (*): 8192 MB

☐ Reserve all guest memory (All locked)

Limit: Unlimited MB

Shares: Normal 81920

Memory Hot Plug: ☐ Enable

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

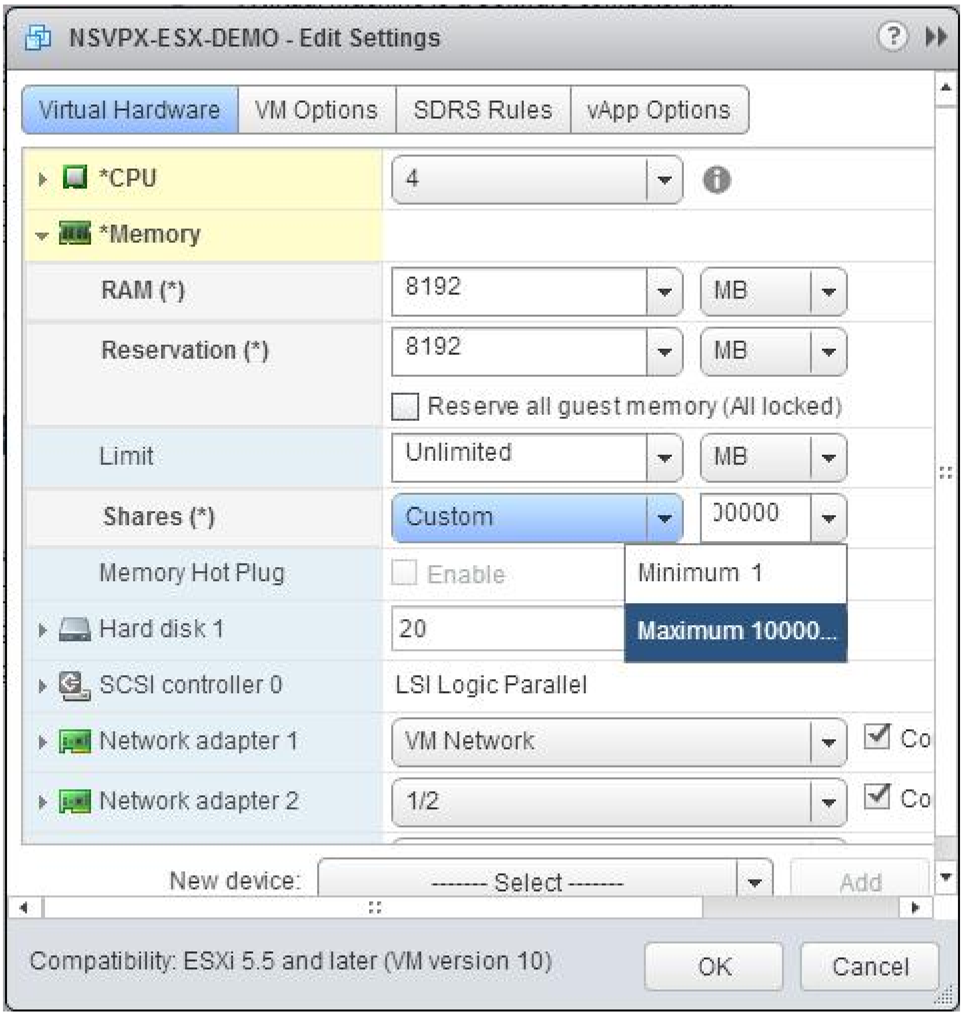
Network adapter 1: VM Network ☑ Co

Network adapter 2: 1/2 ☑ Co

New device: ------- Select ------- Add

Compatibility: ESXi 5.5 and later (VM version 10)

d. In the **Shares** drop-down lists, select **Custom**, and select the number that is shown as the maximum value.


Virtual Hardware
VM Options
SDRS Rules
vApp Options
OK
Cancel
Add

NSVPX-ESX-DEMO - Edit Settings

*CPU: 4

*Memory

RAM (*): 8192 MB

Reservation (*): 8192 MB

☐ Reserve all guest memory (All locked)

Limit: Unlimited MB

Shares (*): Custom)0000

Minimum 1

Maximum 10000...

Memory Hot Plug: ☐ Enable

Hard disk 1: 20

SCSI controller 0: LSI Logic Parallel

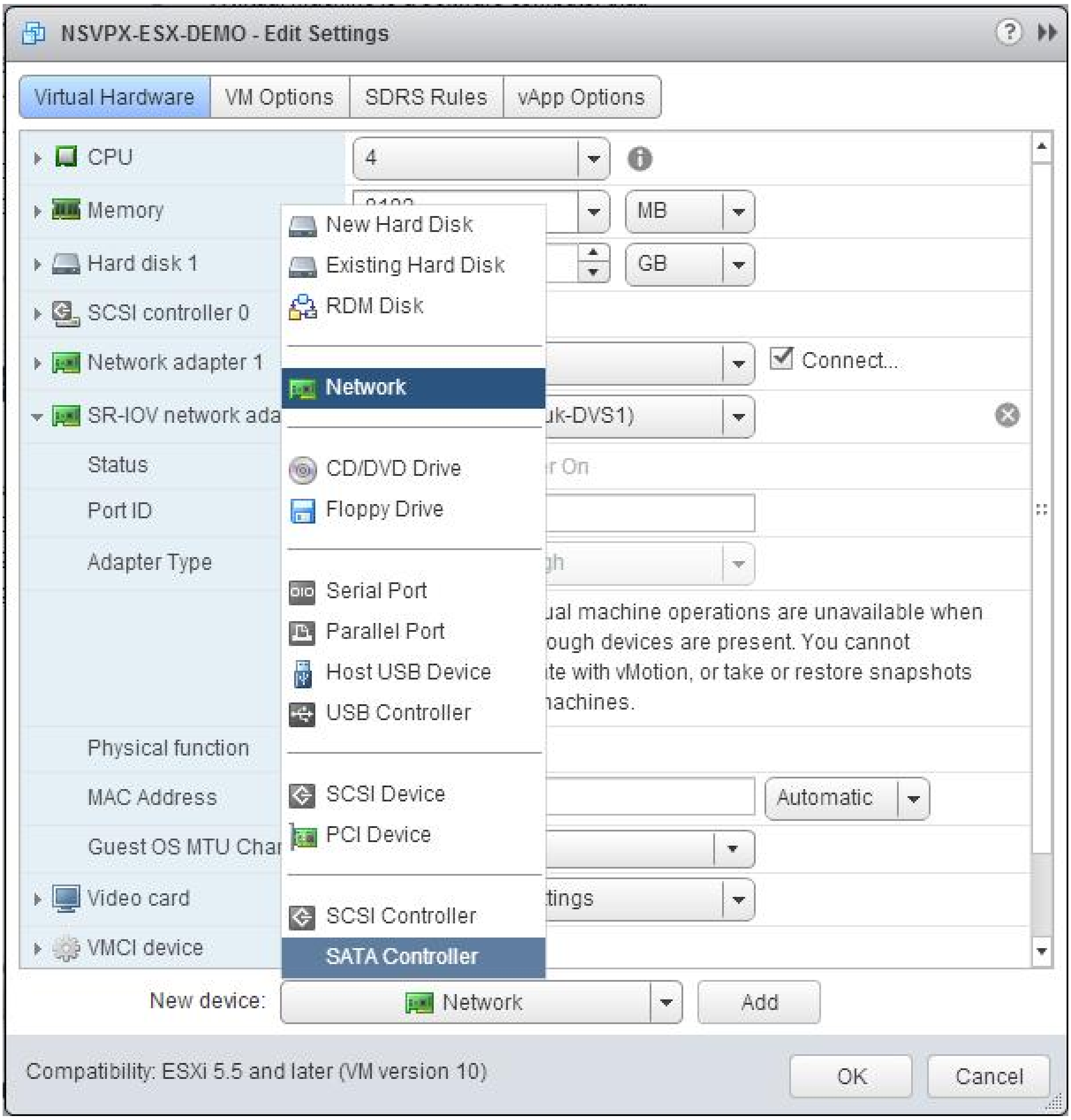
Network adapter 1: VM Network ☑ Co

Network adapter 2: 1/2 ☑ Co

New device: ------- Select -------

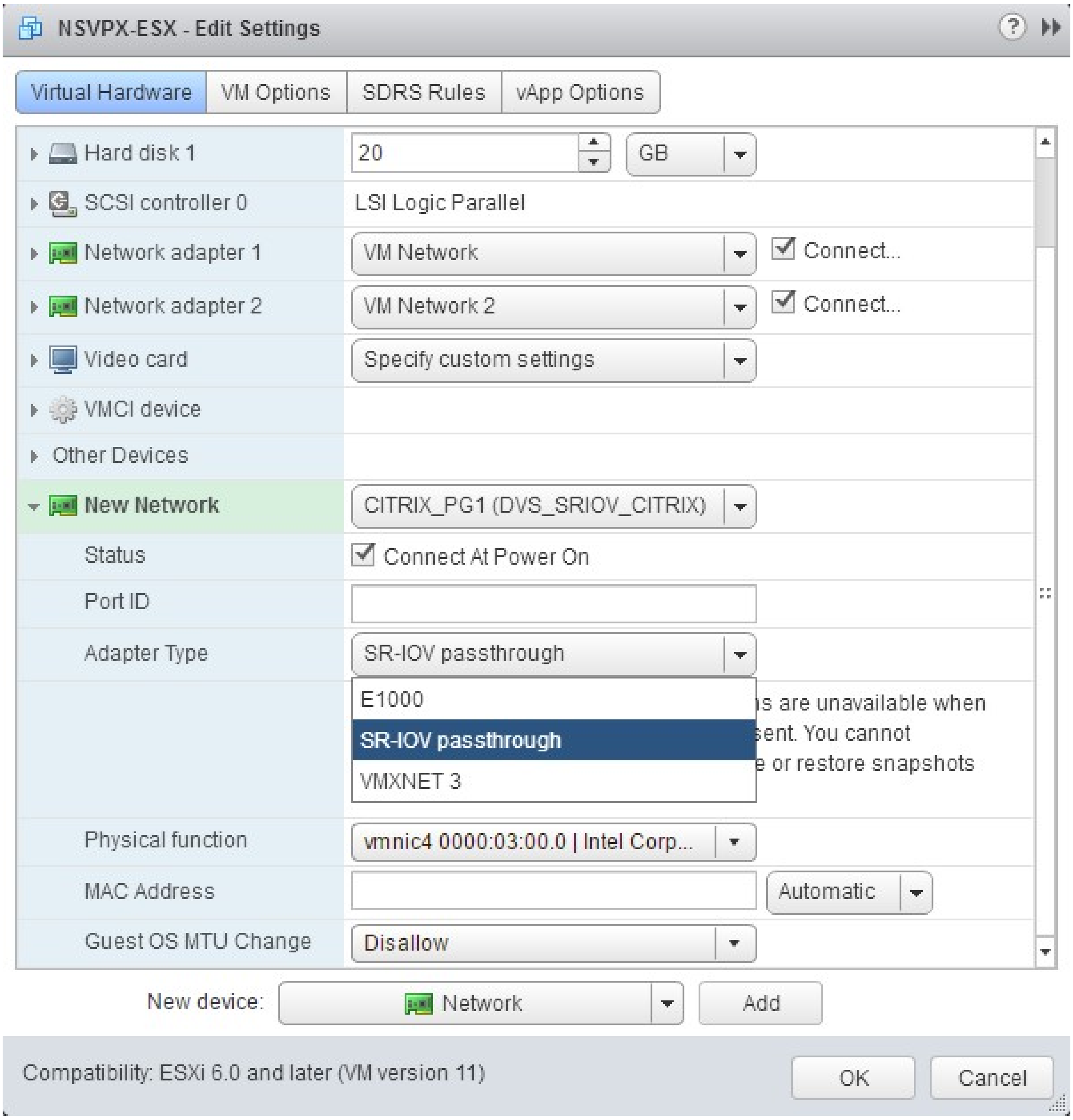
Compatibility: ESXi 5.5 and later (VM version 10)

7. Add an SR-IOV network interface. From the **New device** drop-down list, select **Network** and click **Add**.


NSVPX-ESX-DEMO - Edit Settings
Virtual Hardware
VM Options
SDRS Rules
vApp Options
CPU
4
Memory
MB
Hard disk 1
GB
SCSI controller 0
Network adapter 1
☑ Connect...
SR-IOV network ada
uk-DVS1)
Status
r On
Port ID
Adapter Type
gh
ual machine operations are unavailable when
ough devices are present. You cannot
te with vMotion, or take or restore snapshots
achines.
Physical function
MAC Address
Automatic
Guest OS MTU Char
Video card
tings
VMCI device
New Hard Disk
Existing Hard Disk
RDM Disk
Network
CD/DVD Drive
Floppy Drive
Serial Port
Parallel Port
Host USB Device
USB Controller
SCSI Device
PCI Device
SCSI Controller
SATA Controller
New device:
Network
Add
Compatibility: ESXi 5.5 and later (VM version 10)
OK
Cancel

8. In the **New Network** section. From the drop-down list, select the Portgroup that you created, and do the following:

    a. In the **Adapter Type** drop-down list, select **SR-IOV passthrough**.



NSVPX-ESX - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Network adapter 2: VM Network 2 ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

New Network: CITRIX_PG1 (DVS_SRIOV_CITRIX)

Status: ☑ Connect At Power On

Port ID:

Adapter Type: SR-IOV passthrough

E1000
SR-IOV passthrough
VMXNET 3

s are unavailable when
ent. You cannot
e or restore snapshots

Physical function: vmnic4 0000:03:00.0 | Intel Corp...

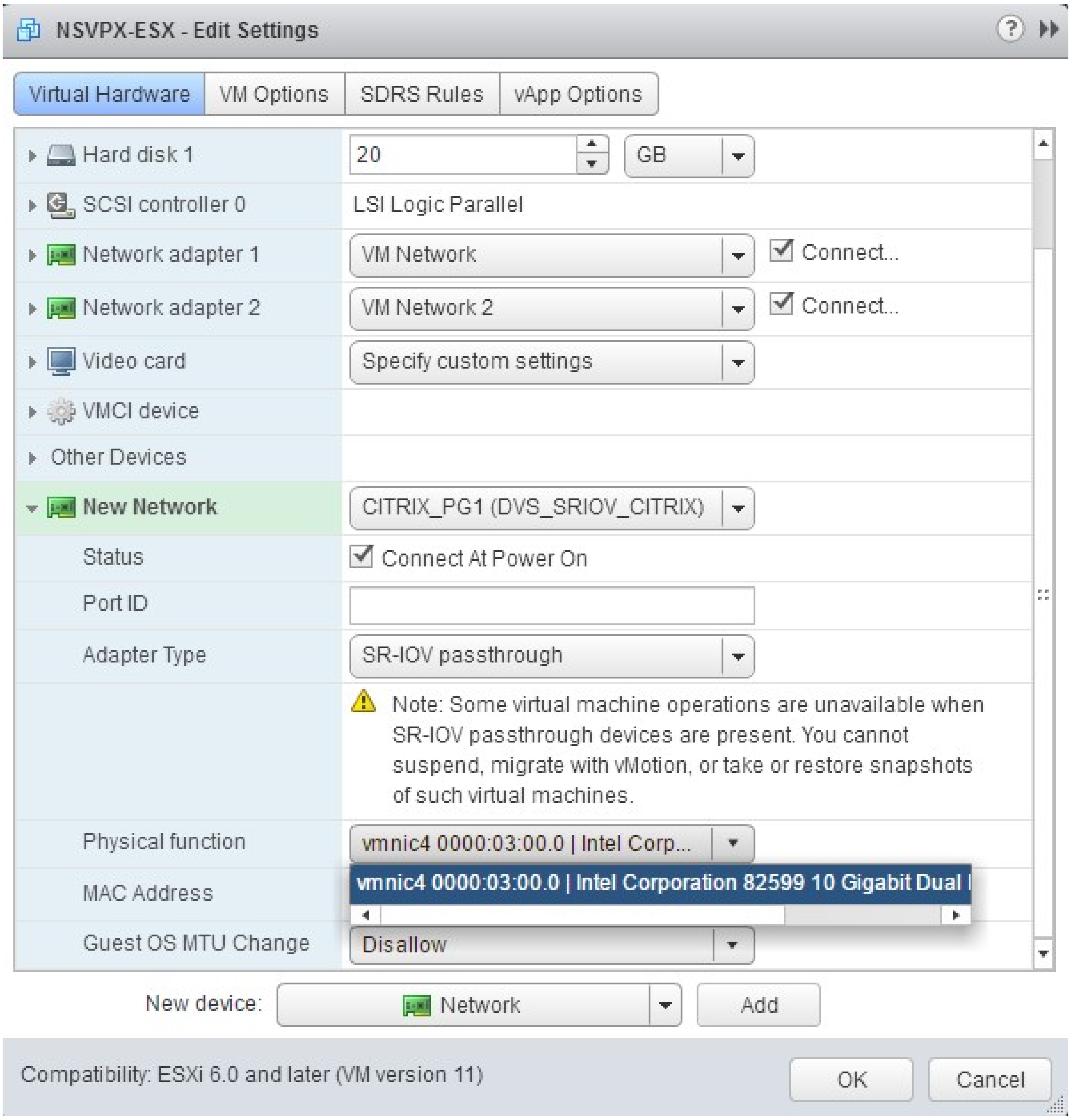
MAC Address: Automatic

Guest OS MTU Change: Disallow

New device: Network

Compatibility: ESXi 6.0 and later (VM version 11)

b. In the **Physical function** drop-down list, select the physical adapter mapped with the Port-group.



NSVPX-ESX - Edit Settings

Virtual Hardware | VM Options | SDRS Rules | vApp Options

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Network adapter 2: VM Network 2 ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

New Network: CITRIX_PG1 (DVS_SRIOV_CITRIX)

Status: ☑ Connect At Power On

Port ID:

Adapter Type: SR-IOV passthrough

Note: Some virtual machine operations are unavailable when SR-IOV passthrough devices are present. You cannot suspend, migrate with vMotion, or take or restore snapshots of such virtual machines.

Physical function: vmnic4 0000:03:00.0 | Intel Corp...

vmnic4 0000:03:00.0 | Intel Corporation 82599 10 Gigabit Dual I

MAC Address

Guest OS MTU Change: Disallow

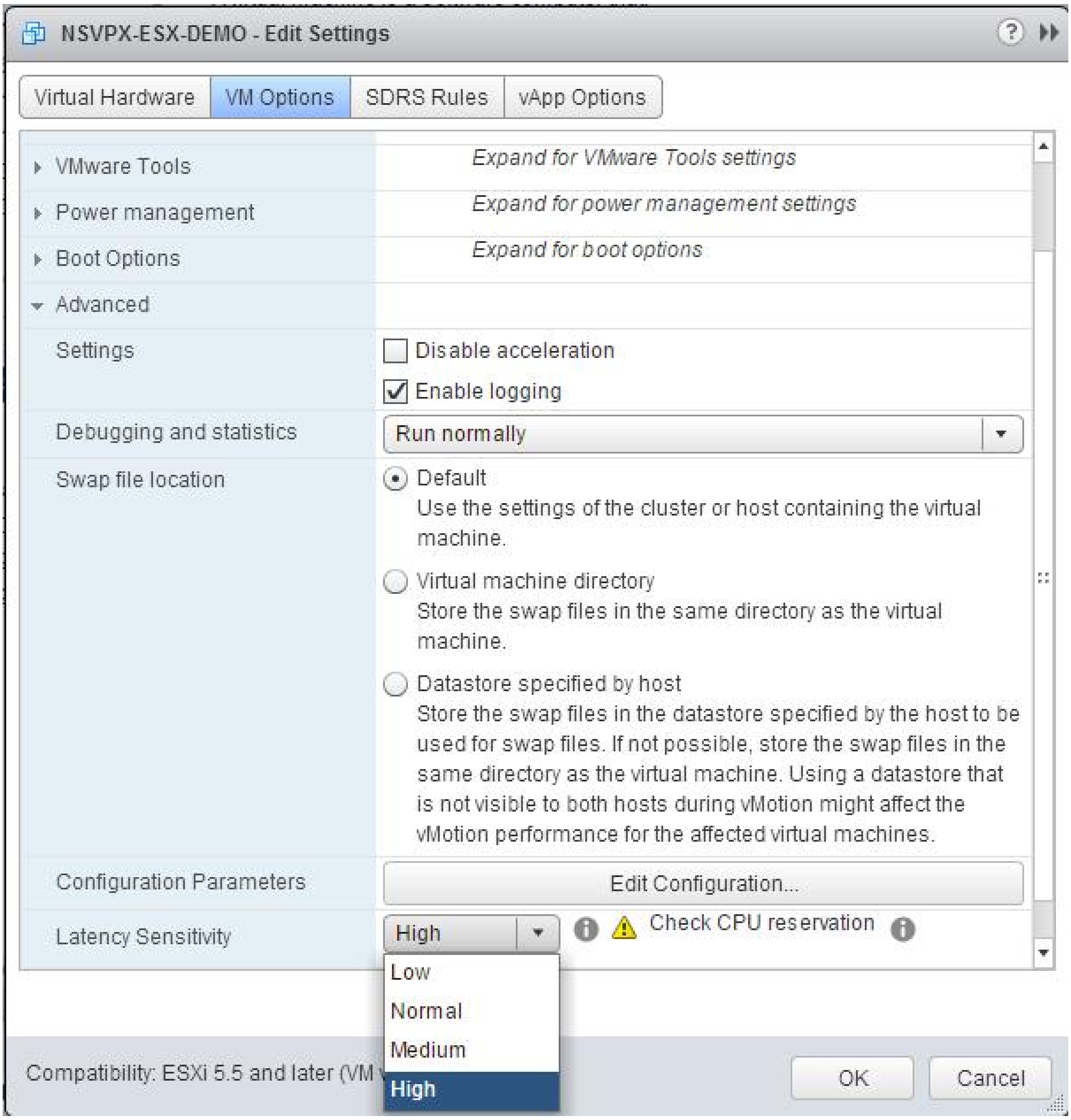
New device: Network Add

Compatibility: ESXi 6.0 and later (VM version 11)

c. In the **Guest OS MTU Change** drop-down list, select **Disallow**.

9. In the **<virtual_appliance> - Edit Settings** dialog box, click the **VM Options** tab.

10. On the **VM Options** tab, select the **Advanced** section. From the **Latency Sensitivity** drop-down list, select **High**.



NSVPX-ESX-DEMO - Edit Settings

Settings: ☐ Disable acceleration ☑ Enable logging

Debugging and statistics: Run normally

Swap file location:
● Default
Use the settings of the cluster or host containing the virtual machine.
○ Virtual machine directory
Store the swap files in the same directory as the virtual machine.
○ Datastore specified by host
Store the swap files in the datastore specified by the host to be used for swap files. If not possible, store the swap files in the same directory as the virtual machine. Using a datastore that is not visible to both hosts during vMotion might affect the vMotion performance for the affected virtual machines.

Latency Sensitivity: High

Check CPU reservation

Compatibility: ESXi 5.5 and later (VM

Virtual Hardware
VM Options
SDRS Rules
vApp Options
VMware Tools
Expand for VMware Tools settings
Power management
Expand for power management settings
Boot Options
Expand for boot options
Advanced
Configuration Parameters
Edit Configuration...
Low
Normal
Medium
High
OK
Cancel

11. Click **OK**.

12. Power on the NetScaler VPX instance.

13. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

```
> show interface summary
```

The output should show all the interfaces that you configured:

```
> show interface summary
```

```
--------------------------------------------------------------------------------
      Interface  MTU        MAC                  Suffix
--------------------------------------------------------------------------------
1     0/1        1500       00:0c:29:1b:81:0b    NetScaler Virtual Interface
2     10/1       1500       00:50:56:9f:0c:6f    Intel 82599 10G VF Interface
3     10/2       1500       00:50:56:9f:5c:1e    Intel 82599 10G VF Interface
4     10/3       1500       00:50:56:9f:02:1b    Intel 82599 10G VF Interface
5     10/4       1500       00:50:56:9f:5a:1d    Intel 82599 10G VF Interface
6     10/5       1500       00:50:56:9f:4e:0b    Intel 82599 10G VF Interface
7     LO/1       1500       00:0c:29:1b:81:0b    Netscaler Loopback interface
 Done
> show inter 10/1
1)      Interface 10/1 (Intel 82599 10G VF Interface) #1
        flags=0xe460 <ENABLED, UP, UP, HAMON, 802.1q>
        MTU=1500, native vlan=55, MAC=00:50:56:9f:0c:6f, uptime 0h21m53s
        Actual: media FIBER, speed 10000, duplex FULL, fctl NONE, throughput 10000
        LLDP Mode: NONE,                 LR Priority: 1024

        RX: Pkts(838020742) Bytes(860888485431) Errs(0) Drops(2527) Stalls(0)
        TX: Pkts(838149954) Bytes(860895860507) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
 Done
```

Migrating the NetScaler VPX from E1000 to SR-IOV or VMXNET3 Network Interfaces

September 21, 2020

You can configure your exiting NetScaler VPX instances that use E1000 network interfaces to use SR-IOV or VMXNET3 network interfaces.

To configure an existing NetScaler VPX instance to use SR-IOV network interfaces, see Configuring NetScaler VPX instances to use Single Root I/O Virtualization (SR-IOV) Network Interface.

To configure an existing NetScaler VPX instance to use VMXNET3 network interfaces, see Configuring NetScaler VPX instances to use VMXNET3 Network Interface.

Configure a NetScaler VPX instance to use PCI passthrough network interface

September 21, 2020

Overview

After you have installed and configured a NetScaler VPX instance on VMware ESX Server, you can use the vSphere Web Client to configure the virtual appliance to use PCI passthrough network interfaces.

The PCI passthrough feature allows a guest virtual machine to directly access physical PCI and PCIe devices connected to a host.

Prerequisites

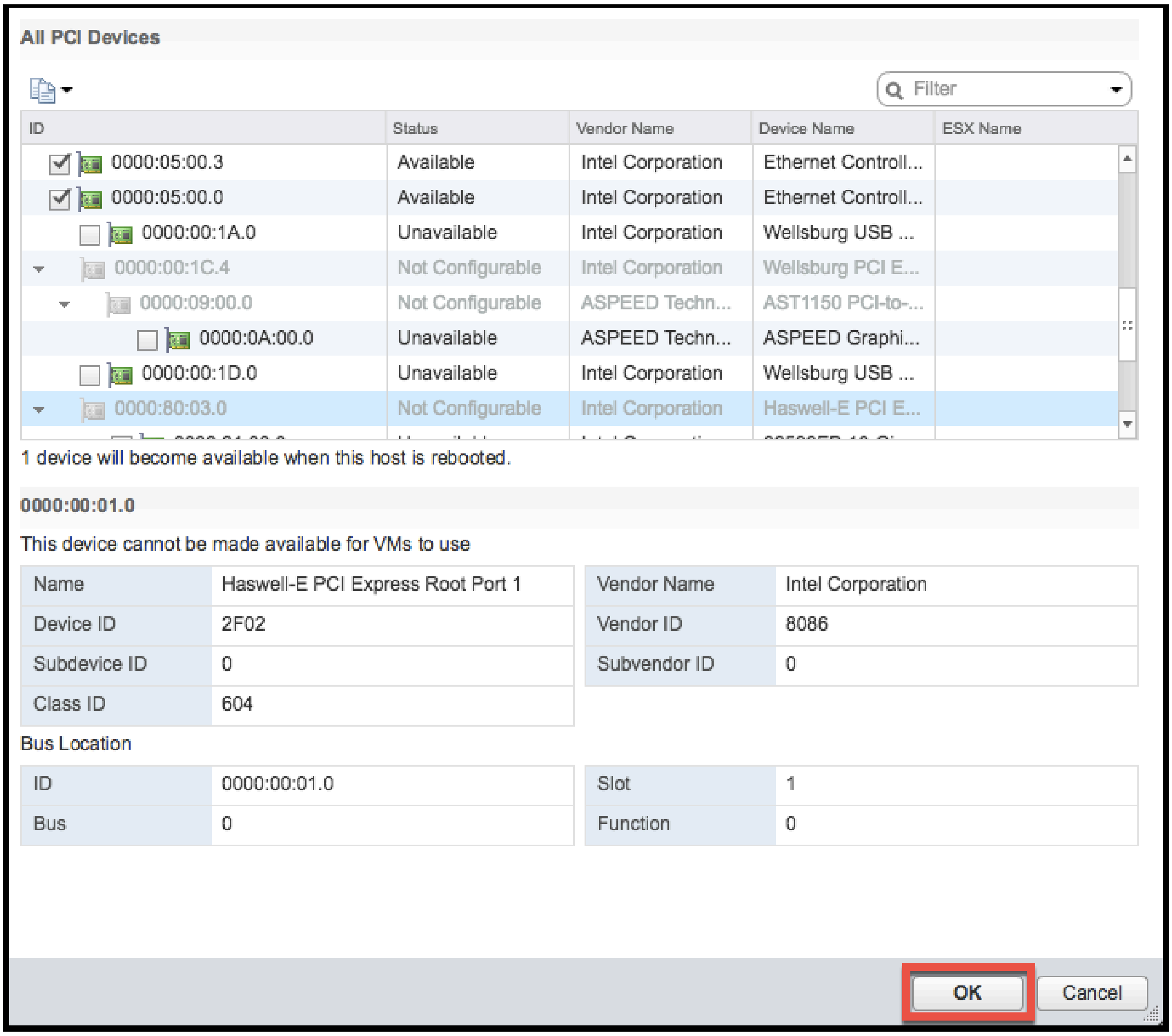
- The firmware version of the Intel XL710 NIC on the host is 5.04.
- A PCI passthrough device connected to and configured on the host
- Supported NICs:
  - Intel X710 10G NIC
  - Intel XL710 Dual Port 40G NIC
  - Intel XL710 Single Port 40G NIC

Configure passthrough devices on a host

Before configuring a passthrough PCI device on a virtual machine, you should configure it on the host machne. Follow these steps to configure passthrough devices on a host.

1. Select the host from the Navigator panel of the vSphere Web Client.
2. Click **Manage** > **Settings** > **PCI Devices**. All available passthrough devices are displayed.
3. Right-click the device that you want to configure and click **Edit**.
4. The **Edit PCI Device Availability** window appears.

5. Select the devices to be used for passthrough and click **OK**.



All PCI Devices

ID	Status	Vendor Name	Device Name	ESX Name
☑ 0000:05:00.3	Available	Intel Corporation	Ethernet Controll...	
☑ 0000:05:00.0	Available	Intel Corporation	Ethernet Controll...	
☐ 0000:00:1A.0	Unavailable	Intel Corporation	Wellsburg USB ...	
0000:00:1C.4	Not Configurable	Intel Corporation	Wellsburg PCI E...	
0000:09:00.0	Not Configurable	ASPEED Techn...	AST1150 PCI-to-...	
☐ 0000:0A:00.0	Unavailable	ASPEED Techn...	ASPEED Graphi...	
☐ 0000:00:1D.0	Unavailable	Intel Corporation	Wellsburg USB ...	
0000:80:03.0	Not Configurable	Intel Corporation	Haswell-E PCI E...	

1 device will become available when this host is rebooted.

0000:00:01.0

This device cannot be made available for VMs to use

Name	Haswell-E PCI Express Root Port 1	Vendor Name	Intel Corporation
Device ID	2F02	Vendor ID	8086
Subdevice ID	0	Subvendor ID	0
Class ID	604		

Bus Location

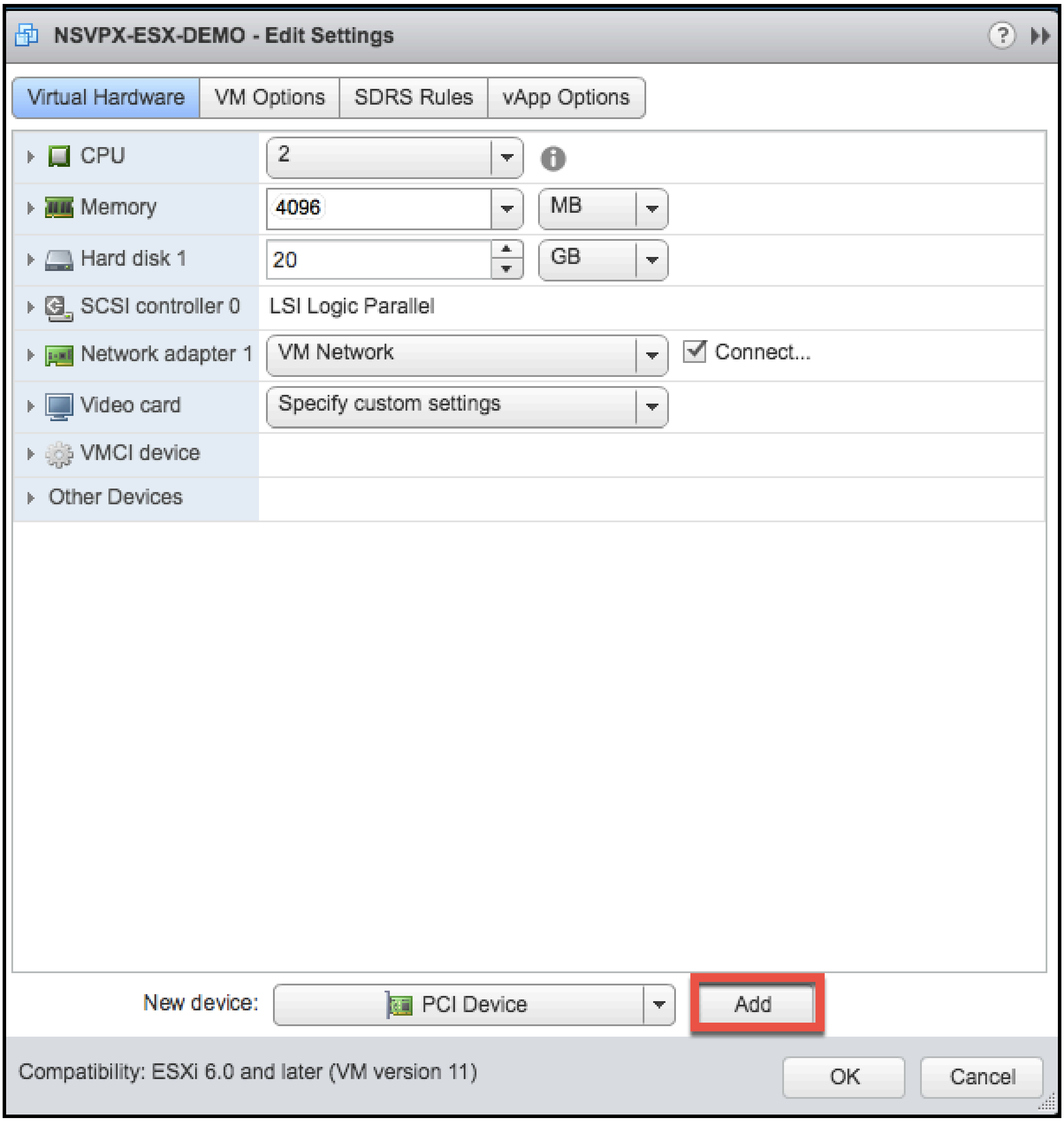
ID	0000:00:01.0	Slot	1
Bus	0	Function	0

6. Restart the host machine.

Configure passthrough devices on a NetScaler VPX instance

Follow these steps to configure a passthrough PCI device on a NetScaler VPX instance.

1. Power off the virtual machine.
2. Right-click the virtual machine and select **Edit Settings**.
3. On the **Virtual Hardware** tab, select **PCI Device** from the **New Device** drop-down menu, and click **Add**.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware
VM Options
SDRS Rules
vApp Options

CPU: 2

Memory: 4096 MB

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

New device: PCI Device

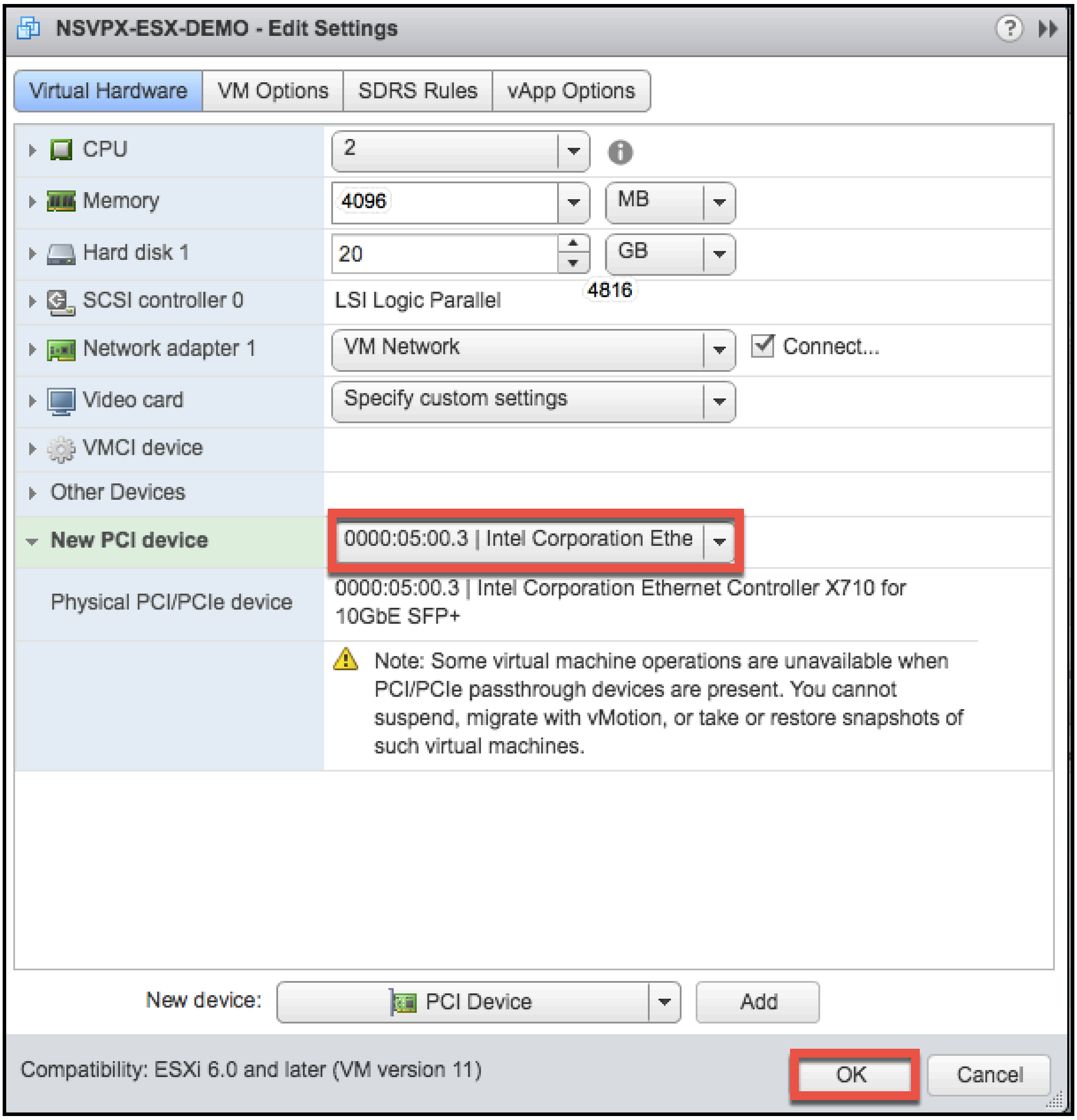
Add

Compatibility: ESXi 6.0 and later (VM version 11)

OK
Cancel

4. Expand **New PCI device** and select the passthrough device to connect to the virtual machine from the drop-down list and click **OK**.

> **Note**
>
> VMXNET3 network interface and PCI Passthrough Network Interface cannot coexist.



NSVPX-ESX-DEMO - Edit Settings

Virtual Hardware | VM Options | SDRS Rules | vApp Options

CPU: 2

Memory: 4096 MB

Hard disk 1: 20 GB

SCSI controller 0: LSI Logic Parallel

Network adapter 1: VM Network ☑ Connect...

Video card: Specify custom settings

VMCI device

Other Devices

New PCI device: 0000:05:00.3 | Intel Corporation Ethe

Physical PCI/PCIe device: 0000:05:00.3 | Intel Corporation Ethernet Controller X710 for 10GbE SFP+

Note: Some virtual machine operations are unavailable when PCI/PCIe passthrough devices are present. You cannot suspend, migrate with vMotion, or take or restore snapshots of such virtual machines.

New device: PCI Device

Compatibility: ESXi 6.0 and later (VM version 11)

6. Power on the guest virtual machine.

You have completed the steps to configuring NetScaler VPX to use PCI passthrough network interfaces.

Install a NetScaler VPX instance on Microsoft Hyper-V server

September 21, 2020

To install Citrix
NetScaler VPX instances on Microsoft Windows Server, you must first install Windows Server, with the

Hyper-V role enabled, on a machine with adequate system resources. While installing the Hyper-V role, be sure to specify the NIC on the server that Hyper-V uses to create the virtual networks. You can reserve some NICs for the host. Use Hyper-V Manager to perform the NetScaler VPX instance installation.

NetScaler VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. It includes the default configuration for elements such as CPU, network interfaces, and hard-disk size and format. After you install NetScaler VPX instance, you can configure the network adapters on virtual appliance, add virtual NICs, and then assign the NetScaler IP address, subnet mask, and gateway, and complete the basic configuration of the virtual appliance.

After the initial configuration of the VPX instance, if you want to upgrade the appliance to the latest software release, see Upgrade a NetScaler VPX standalone appliance

> **Note**
>
> Intermediate System-to-Intermediate System (ISIS) protocol is not supported on the NetScaler VPX virtual appliance hosted on the HyperV-2012 platform.

Prerequisites for installing NetScaler VPX instance on Microsoft servers

Before you begin installing a virtual appliance, do the following:

- Enable the Hyper-V role on Windows Servers. For more information, see http://technet.microsoft.com/en-us/library/ee344837(WS.10).aspx.
- Download the virtual appliance setup files.
- Obtain NetScaler VPX instance license files. For more information about NetScaler VPX instance licenses, see the *NetScaler VPX Licensing Guide* at http://support.citrix.com/article/ctx131110.

Microsoft server hardware requirements

The following table describes the minimum system requirements for Microsoft Servers.

Table 1. Minimum system requirements for Microsoft servers

Component	Requirement
CPU	1.4 GHz 64-bit processor
RAM	3 GB
Disk Space	32 GB or greater

The following table lists the virtual computing resources for each NetScaler VPX instance.

Table 2. Minimum virtual computing resources required for running a NetScaler VPX instance

Component	Requirement
RAM	2 GB
Virtual CPU	2
Disk Space	20 GB
Virtual Network Interfaces	1

Download the NetScaler VPX setup files

The NetScaler VPX instance for Hyper-V is delivered in virtual hard disk (VHD) format. You can download the files from the Citrix website. You need a Citrix account to log in. If you do not have a Citrix account, access the home page at http://www.citrix.com, click **Sign In > My account > Create Citrix Account**, and follow the instructions to create a Citrix account.

To download the NetScaler VPX instance setup files, follow these steps:

1. In a web browser, go to http://www.citrix.com/.
2. Sign in with your user name and password.
3. Click **Downloads**.
4. In **Select a Product** drop-down menu, select **Citrix ADC (NetScaler ADC)**.
5. Under **NetScaler ADC Release X.X > Virtual Appliances**, click **NetScaler VPX Release X.X**
6. Download the compressed file to your server.

Install the NetScaler VPX instance on Microsoft servers

After you have enabled the Hyper-V role on Microsoft Server and extracted the virtual appliance files, you can use Hyper-V Manager to install NetScaler VPX instance. After you import the virtual machine, you need to configure the virtual NICs by associating them to the virtual networks created by Hyper-V.

You can configure a maximum of eight virtual NICs. Even if the physical NIC is DOWN, the virtual appliance assumes that the virtual NIC is UP, because it can still communicate with the other virtual appliances on the same host (server).

> **Note**
>
> You cannot change any settings while the virtual appliance is running. Shut down the virtual appliance and then make changes.

To install NetScaler VPX instance on Microsoft Server by using Hyper-V Manager:

1. To start Hyper-V Manager, click **Start**, point to **Administrative Tools**, and then click **Hyper-V Manager**.
2. In the navigation pane, under **Hyper-V Manage**r, select the server on which you want to install NetScaler VPX instance.
3. On the **Action** menu, click **Import Virtual Machin**e.
4. In the **Import Virtual Machine** dialog box, in **Location**, specify the path of the folder that contains the NetScaler VPX instance software files, and then select **Copy the virtual machine (create a new unique ID)**. This folder is the parent folder that contains the Snapshots, Virtual Hard Disks, and Virtual Machines folders.
5. Note: If you received a compressed file, make sure that you extract the files into a folder before you specify the path to the folder.
6. Click **Import**.
7. Verify that the virtual appliance that you imported is listed under **Virtual Machines**.
8. To install another virtual appliance, repeat steps **2** through **6**.

> **Important**
>
> Make sure that you extract the files to a different folder in step **4**.

Auto-provision a NetScaler VPX instance on Hyper-V

Auto-provisioning of NetScaler VPX instance is optional. If auto-provisioning is not done, the virtual appliance provides an option to configure the IP address and so on.

To auto-provision NetScaler VPX instance on Hyper-V, follow these steps.

1. Create an ISO9660 compliant ISO image using the xml file as depicted in the example. Make sure that the name of the xml file is **userdata**.

<?xml version=”1.0” encoding=”UTF-8” standalone=”no”?>

<Environment xmlns:oe="`http://schemas.dmtf.org/ovf/environment/1`”

xmlns:xsi="`http://www.w3.org/2001/XMLSchema-instance`”

oe:id=””

xmlns="`http://schemas.dmtf.org/ovf/environment/1`”>

<PlatformSection>

<Kind>HYPER-V</Kind>

<Version>2013.1</Version>

<Vendor>CISCO</Vendor>

<Locale>en</Locale>

```
</PlatformSection>

<PropertySection>

<Property oe:key=”com.citrix.netscaler.ovf.version” oe:value=”1.0”/>

<Property oe:key=”com.citrix.netscaler.platform” oe:value=”NS1000V”/>

<Property oe:key=”com.citrix.netscaler.orch_env” oe:value=”cisco-orch-env”/>

<Property oe:key=”com.citrix.netscaler.mgmt.ip” oe:value=”10.102.100.122”/>

<Property oe:key=”com.citrix.netscaler.mgmt.netmask” oe:value=”255.255.255.128”/>

<Property oe:key=”com.citrix.netscaler.mgmt.gateway” oe:value=”10.102.100.67”/></PropertySection>

</Environment>
```

2. Copy the ISO image to hyper-v server.

3. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**. You can also select the virtual appliance and then right click and select **Settings**. The **Settings** window for the selected virtual appliance is displayed.

4. In the **Settings** window, under the hardware section, click on **IDE Controller**.

5. In the right window pane, select **DVD Drive** and click **Add**. The DVD Drive is added under the **IDE Controller** section in the left window pane.

6. Select the **DVD Drive** added in step 5. In the right window pane, select the **Image file** radio button and click on Browse and select the ISO image that you copied on Hyper-V server,
in step 2.

7. Click **Apply**.

> **Note**
>
> The virtual appliance instance comes up in the default IP address, when:
>
> - The DVD drive is attached and the ISO file is not provided.
> - The ISO file does not include the user data file.
> - The user data file name or format is not correct.

To configure virtual NICs on the NetScaler VPX instance, follow these steps:

1. Select the virtual appliance that you imported, and then on the **Action** menu, select **Settings**.
2. In the **Settings for <virtual appliance name>** dialog box, click **Add Hardware** in the left pane.
3. In the right pane, from the list of devices, select **Network Adapter**.
4. Click **Add**.
5. Verify that **Network Adapter (not connected)** appears in the left pane.
6. Select the network adapter in the left pane.

7. In the right pane, from the **Network** drop-down menu, select the virtual network to connect the adapter to.
8. To select the virtual network for additional network adapters that you want to use, repeat steps **6** and **7**.
9. Click **Apply**, and then click **OK**.

To configure the NetScaler VPX instance:

1. Right-click the virtual appliance that you previously installed, and then select **Start**.
2. Access the console by double-clicking the virtual appliance.
3. Type the NetScaler IP address, subnet mask, and gateway for your virtual appliance.

You have completed the basic configuration of your virtual appliance. Type the IP address in a Web browser to access the virtual appliance.

> **Note**
>
> You can also use virtual machine (VM) template to provision NetScaler VPX instance using SCVMM.
>
> If you use Microsoft Hyper-V NIC teaming solution with NetScaler VPX instances, see article CTX224494 for more information.

Install a NetScaler VPX instance on Linux-KVM platform

September 21, 2020

To set up a NetScaler VPX for the Linux-KVM platform, you can use the graphical Virtual Machine Manager (Virt-Manager) application. If you prefer the Linux-KVM command line, you can use the virsh program.

The host Linux operating system must be installed on suitable hardware by using virtualization tools such as KVM Module and QEMU. The number of virtual machines (VMs) that can be deployed on the hypervisor depends on the application requirement and the chosen hardware.

After you provision a NetScaler VPX instance, you can add additional interfaces.

Limitations and usage guidelines

General recommendations

To avoid unpredictable behavior, apply the following recommendations:

- Do not change the MTU of the vnet interface associated with the VPX VM. Shut down the VPX VM before modifying any configuration parameters, such as Interface modes or CPU.
- Do not force a shutdown of the VPX VM. That is, do not use the **Force off** command.

- Any configurations done on the host Linux might or might not be persistent, depending on your Linux distribution settings. You can choose to make these configurations persistent to ensure consistent behavior across reboots of host Linux operating system.
- The NetScaler package has to be unique for each of the NetScaler VPX instance provisioned.

Limitations

- Live migration of a VPX instance that runs on KVM is not supported.

Prerequisites for installing a NetScaler VPX instance on Linux-KVM platform

March 30, 2021

Check the minimum system requirements for a Linux-KVM serves running a NetScaler VPX instance.

CPU requirement:

- 64-bit x86 processors with the hardware virtualization features included in the AMD-V and Intel VT-X processors.

To test whether your CPU supports Linux host, enter the following command at the host Linux shell prompt:

```
*.egrep'^flags.*(vmx|svm)'/proc/cpuinfo*
<!--NeedCopy-->
```

If the BIOS settings for the above extension are disabled, you must enable them in BIOS.

- Provide at least 2 CPU cores to Host Linux.
- There is no specific recommendation for processor speed, but higher the speed, the better the performance of the VM application.

Memory (RAM) requirement:

Minimum 4 GB for the host Linux kernel. Add additional memory as required by the VMs.

Hard disk requirement:

Calculate the space for Host Linux kernel and VM requirements. A single NetScaler VPX VM requires 20 GB of disk space.

Software requirements

The Host kernel used must be a 64-bit Linux kernel, release 2.6.20 or later, with all virtualization tools. Citrix recommends newer kernels, such as 3.6.11-4 and later.

Many Linux distributions such as Red Hat, Centos, and Fedora, have tested kernel versions and associated virtualization tools.

Guest VM hardware requirements

NetScaler VPX supports IDE and virtIO hard disk type. The Hard Disk Type has been configured in the XML file, which is a part of the NetScaler package.

Networking requirements

NetScaler VPX supports virtIO para-virtualized, SR-IOV, and PCI Passthrough network interfaces.

For more information about the supported network interfaces, see:

- Provision the NetScaler VPX instance by using the Virtual Machine Manager
- Configure a NetScaler VPX instance to use SR-IOV network interfaces
- Configure a NetScaler VPX instance to use PCI passthrough network interfaces

Source Interface and Modes

The source device type can be either Bridge or MacVTap. In case of MacVTap, four modes are possible - VEPA, Bridge, Private and Pass-through.
Check the types of interfaces that you can use and the supported traffic types, as given below.

Bridge:

- Linux Bridge.
- Ebtables and iptables settings on host Linux might filter the traffic on the bridge if you do not choose the correct setting or disable IPtable services.

MacVTap (VEPA mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.
- Inter-VM communication using the same
- lower device is possible only if upstream or downstream switch supports VEPA mode.

MacVTap (private mode):

- Better performance than a bridge.
- Interfaces from the same lower device can be shared across the VMs.

- Inter-VM communication using the same lower device is not possible.

MacVTap (bridge mode):

- Better as compared to bridge.
- Interfaces out of same lower device can be shared across the VMs.
- Inter-VM communication using the same lower device is possible, if lower device link is UP.

MacVTap (Pass-through mode):

- Better as compared to bridge.
- Interfaces out of same lower device cannot be shared across the VMs.
- Only one VM can use the lower device.

> Note: For best performance by the VPX instance, ensure that the gro and lro capabilities are switched off on the source interfaces.

Properties of source interfaces

Make sure that you switch off the generic-receive-offload (gro) and large-receive-offload (lro) capabilities of the source interfaces. To switch off the gro and lro capabilities, run the following commands at the host Linux shell prompt.

ethtool -K eth6 gro off

ethool -K eth6 lro off

Example:

```
    [root@localhost ~]# ethtool -K eth6

                    Offload parameters for eth6:

                                    rx-checksumming: on

                                    tx-checksumming: on

                    scatter-gather: on

                    tcp-segmentation-offload: on

                    udp-fragmentation-offload: off

                    generic-segmentation-offload: on

                    generic-receive-offload: off

```

```
                     large-receive-offload: off

                     rx-vlan-offload: on

                     tx-vlan-offload: on

                     ntuple-filters: off

                     receive-hashing: on

    [root@localhost ~]#
<!--NeedCopy-->
```

Example:

If the host Linux bridge is used as a source device, as in the following example, gro and lro capabilities must be switched off on the vnet interfaces, which are the virtual interfaces connecting the host to the guest VMs.

```
    [root@localhost ~]# brctl show eth6_br

    bridge name     bridge  id               STP enabled interfaces

    eth6_br         8000.00e0ed1861ae          no          eth6

                                                          vnet0

                                                          vnet2

    [root@localhost ~]#
<!--NeedCopy-->
```

In the above example, the two virtual interfaces are derived from the eth6_br and are represented as vnet0 and vnet2. Run the following commands to switch off gro and lro capabilities on these interfaces.

```
    ethtool -K vnet0 gro off
            ethtool -K vnet2 gro off
            ethtool -K vnet0 lro off
                     ethtool -K vnet2 lro off
<!--NeedCopy-->
```

Promiscuous mode

The promiscuos mode has to be enabled for the following features to work:

- L2 mode
- Multicast traffic processing
- Broadcast
- IPV6 traffic
- Virtual MAC
- Dynamic routing

Use the following command to enable the promicuous mode.

```
    [root@localhost ~]# ifconfig eth6 promisc
    [root@localhost ~]# ifconfig eth6
    eth6       Link encap:Ethernet  HWaddr 78:2b:cb:51:54:a3
              inet6 addr: fe80::7a2b:cbff:fe51:54a3/64 Scope:Link
              UP BROADCAST RUNNING PROMISC MULTICAST  MTU:9000  Metric:1
              RX packets:142961 errors:0 dropped:0 overruns:0 frame:0
              TX packets:2895843 errors:0 dropped:0 overruns:0 carrier:0
              collisions:0 txqueuelen:1000
              RX bytes:14330008 (14.3 MB)  TX bytes:1019416071 (1.0 GB)

    [root@localhost ~]#
<!--NeedCopy-->
```

Module required

For better network performance, make sure the vhost_net module is present in the Linux host. To check the existence of vhost_net module, run the following command on the Linux host :

```
lsmod | grep "vhost\_net"
<!--NeedCopy-->
```

If vhost_net is not yet running, enter the following command to run it:

```
modprobe vhost\_net
<!--NeedCopy-->
```

Provision the NetScaler VPX instance by using OpenStack

September 21, 2020

You can provision a NetScaler VPX instance in an Openstack environment either by using the Nova boot command (OpenStack CLI) or Horizon (OpenStack dashboard) .

Provisioning a VPX instance, optionally involves using data from the config drive. Config drive is a special configuration drive that attaches to the instance as a CD-ROM device when it boots. This configuration drive can be used to pass networking configuration such as management IP address, network mask, default gateway, and to inject customer scripts.

In a NetScaler appliance, the default authentication mechanism is password based. Now, SSH keypair authentication mechanism is supported for NetScaler VPX instances on OpenStack environment.

The key-pair (public key and private key) needs to be generated before using Public Key Cryptography mechanism. You can use different mechanisms, such as Horizon, Puttygen.exe for Windows, and ssh-keygen for Linux environment, to generate the key pair. Refer to online documentation of respective mechanisms for more information about generating key pair.

Once a key pair is available, copy the private key to a secure location to which authorized persons will have access. In OpenStack, public key can be deployed on a VPX instance by using Horizon or Nova boot command. When a VPX instance is provisioned by using OpenStack, it first detects that the instance is booting in an OpenStack environment by reading a specific BIOS string. This string is "OpenStack Foundation" and for Red Hat Linux distributions it is stored in /etc/nova/release. This is a standard mechanism that is available in all OpenStack implementations based on KVM hypervisor platform. The drive should have a specific OpenStack label.

If the config drive is detected, the instance attempts to read the network configuration, custom scripts, and SSH key pair if provided.

Userdata file

The NetScaler VPX instance uses a customized OVF file, also known as userdata file, to inject network configuration, custom scripts. This file is provided as part of config drive. Here is an example of a customized OVF file.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<Environment xmlns:oe="http://schemas.dmtf.org/ovf/environment/1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
oe:id=""
xmlns="http://schemas.dmtf.org/ovf/environment/1"
xmlns:cs="http://schemas.citrix.com/openstack">
<PlatformSection>
```

```
<Kind></Kind>
<Version>2016.1</Version>
<Vendor>VPX</Vendor>
<Locale>en</Locale>
</PlatformSection>
<PropertySection>
<Property oe:key="com.citrix.netscaler.ovf.version" oe:value="1.0"/>
<Property oe:key="com.citrix.netscaler.platform" oe:value="NSVPX"/>
<Property oe:key="com.citrix.netscaler.orch_env" oe:value="openstack-orch-env"/>
<Property oe:key="com.citrix.netscaler.mgmt.ip" oe:value="10.1.2.22"/>
<Property oe:key="com.citrix.netscaler.mgmt.netmask" oe:value="255.255.255.0"/>
<Property oe:key="com.citrix.netscaler.mgmt.gateway" oe:value="10.1.2.1"/>
</PropertySection>
 <cs:ScriptSection>
   <cs:Version>1.0</cs:Version>
     <ScriptSettingSection xmlns="http://schemas.citrix.com/openstack" xmlns:i="http://www.w3.org/2001/XMLSchema-instance">
         <Scripts>
           <Script>
                 <Type>shell</Type>
                 <Parameter>X Y</Parameter>
                <Parameter>Z</Parameter>
                <BootScript>before</BootScript>
                  <Text>
                        #!/bin/bash
                        echo "Hi, how are you" $1 $2 >> /var/sample.txt
                  </Text>
           </Script>
           <Script>
                 <Type>python</Type>
                 <BootScript>after</BootScript>
                  <Text>
                       #!/bin/python
 print("Hello");
                  </Text>
           </Script>
   <Script>
                 <Type>perl</Type>
                 <BootScript>before</BootScript>
                  <Text>
                       !/usr/bin/perl
 my $name = "VPX";
 print "Hello, World $name !\n" ;
                  </Text>
           </Script>
           <Script>
                <Type>nscli</Type>
                <BootScript>after</BootScript>
                <Text>
                 add vlan 33
 bind vlan 33 -ifnum 1/2
                </Text>
           </Script>
         </Scripts>
     </ScriptSettingSection>
 </cs:ScriptSection>
</Environment>
```

In the OVF file above "PropertySection" is used for NetScaler networking configuration while <cs:ScriptSection> is used to enclose all scripts. <Scripts></Scripts> tags are used to bundle all scripts together. Each script is defined in between <Script> </Script> tags. Each script tag has following fields/tags:

a) <Type>: Specifies value for script type. Possible values: Shell/Perl/Python/NSLCI (for NetScaler CLI scripts)

b) <Parameter>: Provides parameters to the script. Each script can have multiple <Parameter> tags.

c) <BootScript>: Specifies script execution point. Possible values for this tag: before/after. "before" specifies script will be executed before PE comes up. "after" specifies that the script will be executed after PE comes up.

d) <Text>: Pastes content of a script.

> **Note**
>
> Currently the VPX instance does not take care of sanitization of scripts. As an administrator, you should check the validity of the script.
>
> Not all sections need to be present. Use an empty "PropertySection" to only define scripts to execute on first boot or an empty <cs:ScriptSection> to only define networking configuration.
>
> After the required sections of the OVF file (userdata file) are populated, use that file to provision the VPX instance.

Network configuration

As part of network configuration, the VPX instance reads:

- Management IP address
- Network mask
- Default gateway

After the parameters are successfully read, they are populated in the NetScaler configuration, to allow managing the instance remotely. If the parameters are not read successfully or the config drive is not available, the instance transitions to the default behavior, which is:

- The instance attempts to retrieve the IP address information from DHCP.
- If DHCP fails or times-out, the instance comes up with default network configuration (192.168.100.1/16).

Customer script

The VPX instance allows to execute a custom script during initial provisioning. The appliance supports script of type Shell, Perl, Python, and NetScaler CLI commands.

SSH key pair authentication

The VPX instance copies public key, available within the configuration drive as part of instance meta data, into its "authorized_keys" file. This allows the user to access the instance with private key.

> **Note**
>
> When an SSH key is provided, the default credentials (nsroot/nsroot) no longer work. If password-based access is needed, log on with the respective SSH private key and manually set a password.

Before you begin

Before you provision a VPX instance on OpenStack environment, extract the .qcow2 file from the .tgz file and build

an OpenStack image from the qcow2 image. Follow these steps:

1. Extract the. qcow2 file from the .tqz file by typing the following command.

```
tar xvzf <TAR file>

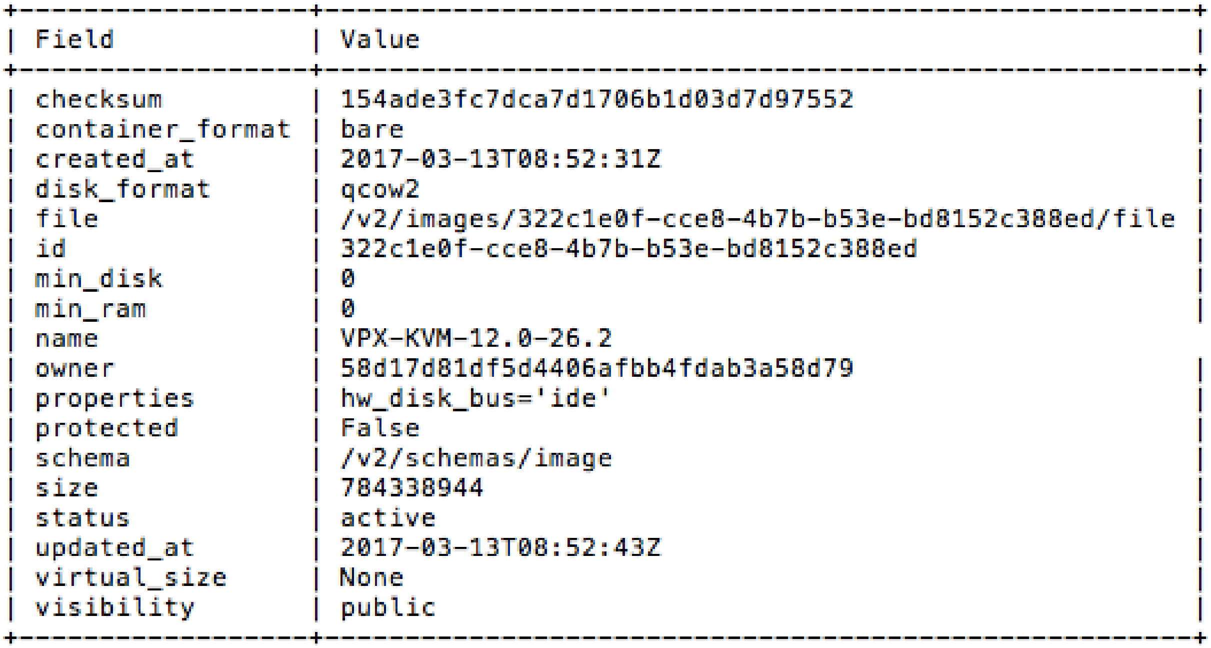
tar xvzf <NSVPX-KVM-12.0-26.2_nc.tgz>
NSVPX-KVM.xml
NSVPX-KVM-12.0-26.2_nc.qcow2
```

2. Build an OpenStack image using the .qcoz2 file extracted in step 1 by typing the following command.

```
openstack image create --container-format bare --property hw_disk_bus=ide --disk-format qcow2 --file <path to qcow2 file> --public <name of the OpenStack image>

glance image-create --name="NS-VPX-12-0-26-2" --property hw_disk_bus=ide --ispublic=
true --container-format=bare --disk-format=qcow2< NSVPX-KVM-12.0-26.2_nc.qcow2
```

Figure 1: The following illustration provides a sample output for the glance image-create command.



```
+------------------+------------------------------------------------------+
| Field            | Value                                                |
+------------------+------------------------------------------------------+
| checksum         | 154ade3fc7dca7d1706b1d03d7d97552                     |
| container_format | bare                                                 |
| created_at       | 2017-03-13T08:52:31Z                                 |
| disk_format      | qcow2                                                |
| file             | /v2/images/322c1e0f-cce8-4b7b-b53e-bd8152c388ed/file |
| id               | 322c1e0f-cce8-4b7b-b53e-bd8152c388ed                 |
| min_disk         | 0                                                    |
| min_ram          | 0                                                    |
| name             | VPX-KVM-12.0-26.2
| owner            | 58d17d81df5d4406afbb4fdab3a58d79                     |
| properties       | hw_disk_bus='ide'                                    |
| protected        | False                                                |
| schema           | /v2/schemas/image                                    |
| size             | 784338944                                            |
| status           | active                                               |
| updated_at       | 2017-03-13T08:52:43Z                                 |
| virtual_size     | None                                                 |
| visibility       | public                                               |
+------------------+------------------------------------------------------+
```

Provisioning the VPX instance

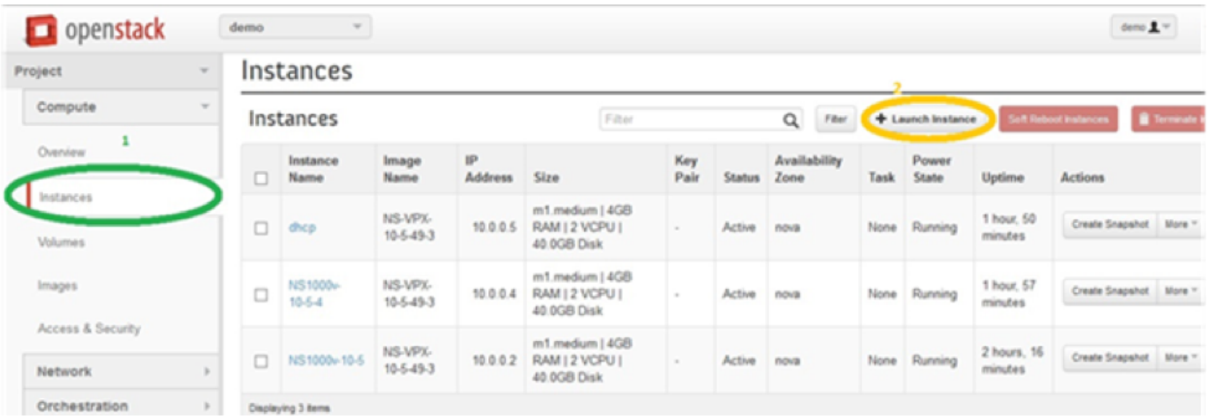
You can provision a VPX instance in two ways by using one of the options:

- Horizon (OpenStack dashboard)
- Nova boot command (OpenStack CLI)

Provision a VPX instance by using the OpenStack dashboard

Follow these steps to provision the VPX instance by using Horizon:

1. Log on to the OpenStack dashboard.
2. In the Project panel on the left hand side of the dashboard, select **Instances**.
3. In the Instances panel, click **Launch Instance** to open the Instance Launching wizard.

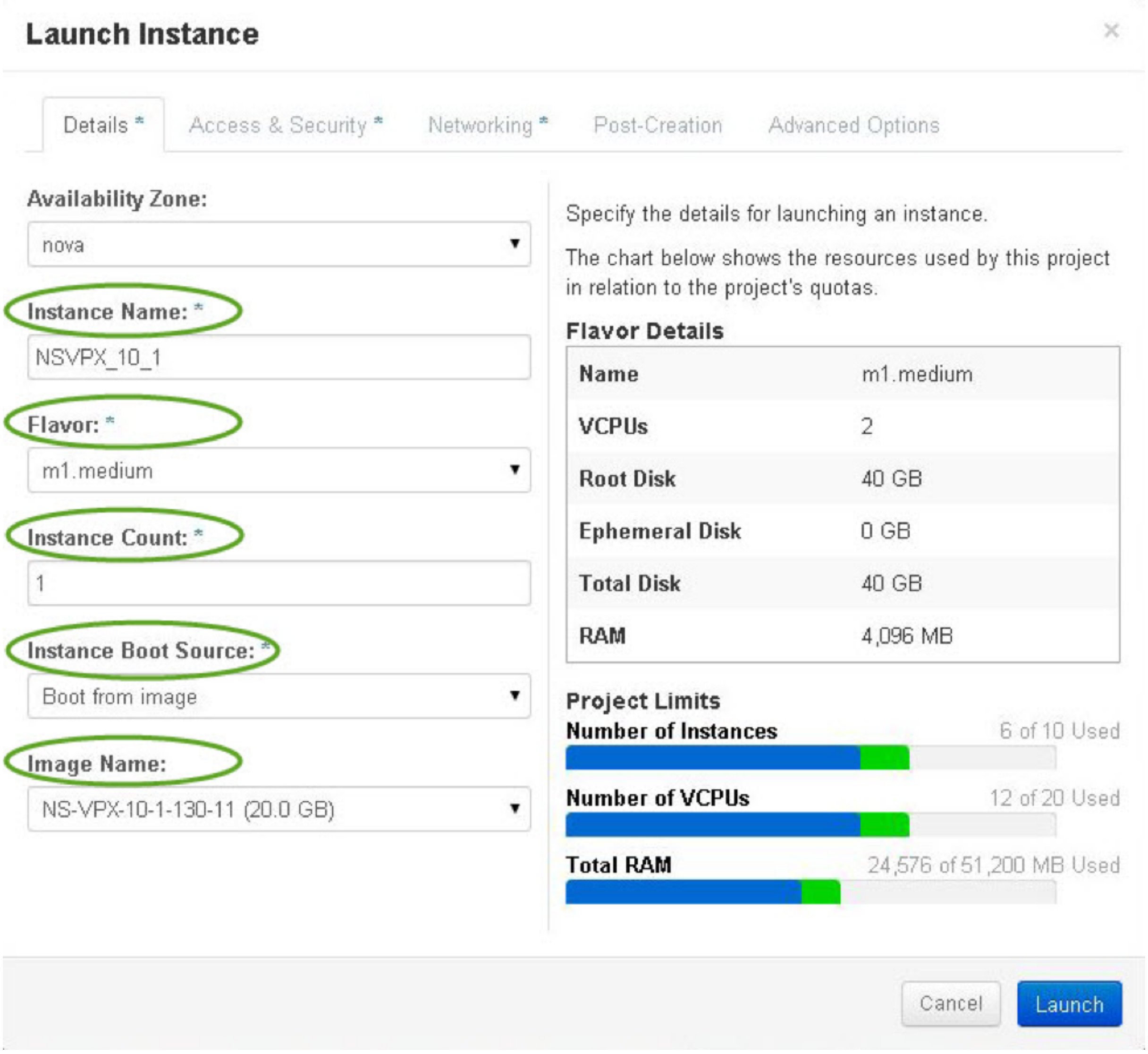

openstack
demo
Project
Compute
1
Overview
Instances
Volumes
Images
Access & Security
Network
Orchestration
Instances
Instances
Filter
Filter
2
Launch Instance
Soft Reboot Instances
Terminate I
Create Snapshot
More

☐	Instance Name	Image Name	IP Address	Size	Key Pair	Status	Availability Zone	Task	Power State	Uptime	Actions
☐	dhcp	NS-VPX-10-5-49-3	10.0.0.5	m1.medium | 4GB RAM | 2 VCPU | 40.0GB Disk	-	Active	nova	None	Running	1 hour, 50 minutes	Create Snapshot More
☐	NS1000v-10-5-4	NS-VPX-10-5-49-3	10.0.0.4	m1.medium | 4GB RAM | 2 VCPU | 40.0GB Disk	-	Active	nova	None	Running	1 hour, 57 minutes	Create Snapshot More
☐	NS1000v-10-5	NS-VPX-10-5-49-3	10.0.0.2	m1.medium | 4GB RAM | 2 VCPU | 40.0GB Disk	-	Active	nova	None	Running	2 hours, 16 minutes	Create Snapshot More

Displaying 3 items

4. In the Launch Instance wizard, fill in the details, like:

   1. Instance Name
   2. Instance Flavor
   3. Instance Count
   4. Instance Boot Source
   5. Image Name



Availability Zone: nova

Instance Name: * NSVPX_10_1

Flavor: * m1.medium

Instance Count: * 1

Instance Boot Source: * Boot from image

Image Name: NS-VPX-10-1-130-11 (20.0 GB)

Specify the details for launching an instance.

The chart below shows the resources used by this project in relation to the project's quotas.

Flavor Details

Name	m1.medium
VCPUs	2
Root Disk	40 GB
Ephemeral Disk	0 GB
Total Disk	40 GB
RAM	4,096 MB

Project Limits

Number of Instances 6 of 10 Used

Number of VCPUs 12 of 20 Used

Total RAM 24,576 of 51,200 MB Used

5. Deploy a new key pair or an existing key pair through Horizon by completing the following steps:

a) If you don't have an existing key pair, create the key by using any existing mechanisms. If you've an existing key, skip this step.

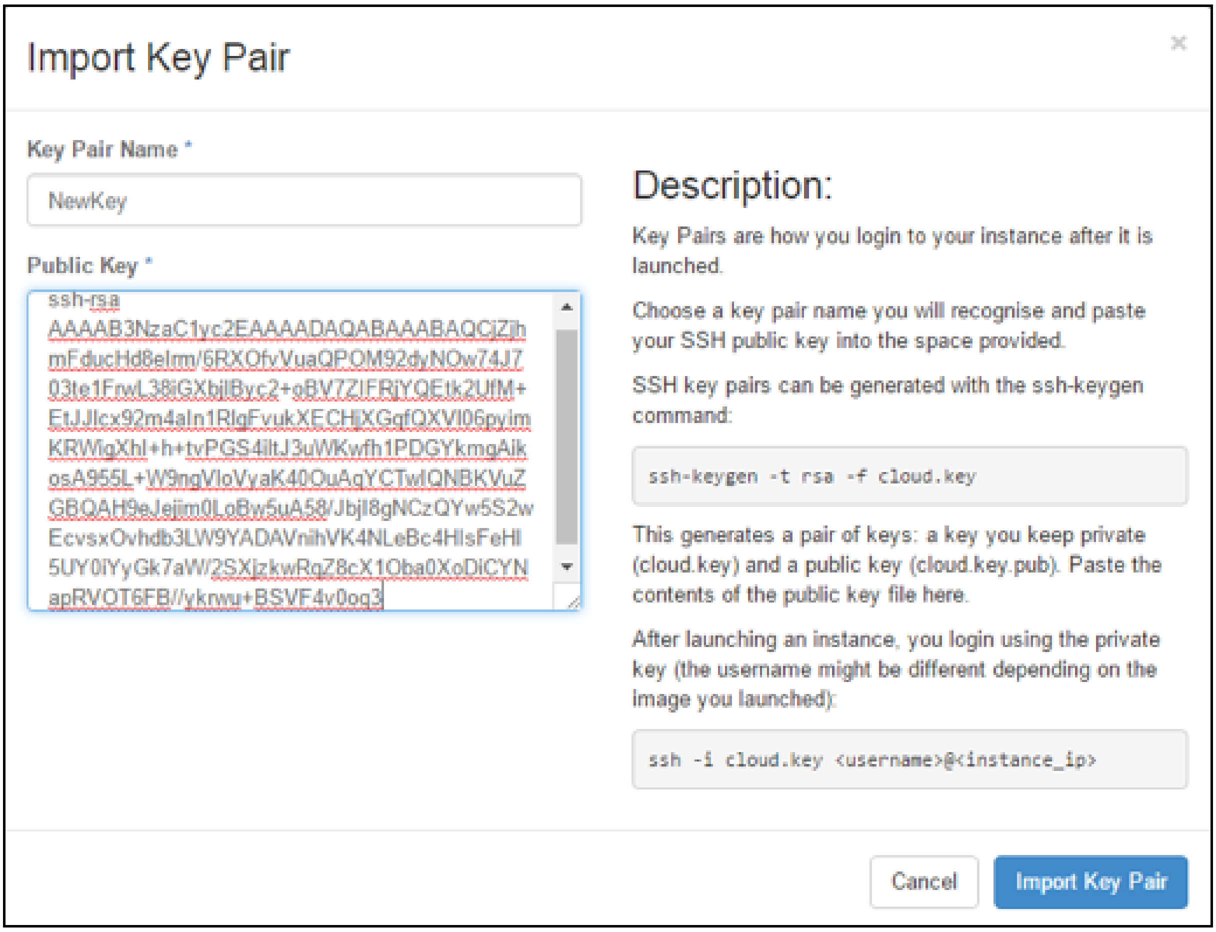
b) Copy the content of public key.

c) Go to **Horizon > Instances > Create New Instances**.

d) Click **Access & Security**.

e) Click the + sign next to the **Key Pair** drop-down menu and provide values for shown parameters.

f) Paste public key content in *Public key* box, give a name to the key and click **Import Key Pair**.


Import Key Pair
×
Key Pair Name *
Public Key *
Description:
Cancel
Import Key Pair

Key Pair Name: NewKey

Public Key:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCjZjh
mFducHd8elmm/6RXOfvVuaQPOM92dyNOw74J7
03te1FnwL38iGXbjlByc2+oBV7ZIFRjYQEtk2UfM+
EtJJlcx92m4aln1RlgFvukXECHjXGqfQXVI06pyim
KRWigXhl+h+tvPGS4iltJ3uWKwfh1PDGYkmgAik
osA955L+W9ngVloVyaK40OuAqYCTwIQNBKVuZ
GBQAH9eJejim0LoBw5uA58/Jbjl8gNCzQYw5S2w
EcvsxOvhdb3LW9YADAVnihVK4NLeBc4HIsFeHl
5UY0iYyGk7aW/2SXjzkwRqZ8cX1Oba0XoDiCYN
apRVOT6FB//yknwu+BSVF4v0oq3
```

Key Pairs are how you login to your instance after it is launched.

Choose a key pair name you will recognise and paste your SSH public key into the space provided.

SSH key pairs can be generated with the ssh-keygen command:

```
ssh-keygen -t rsa -f cloud.key
```

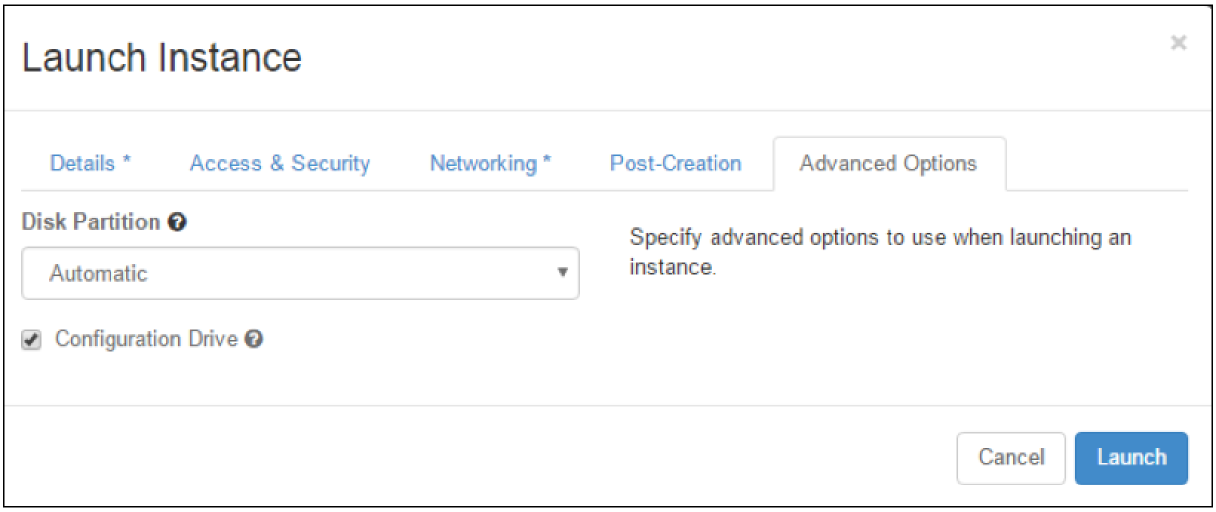
This generates a pair of keys: a key you keep private (cloud.key) and a public key (cloud.key.pub). Paste the contents of the public key file here.

After launching an instance, you login using the private key (the username might be different depending on the image you launched):

```
ssh -i cloud.key <username>@<instance_ip>
```

6. Click the **Post Creation** tab in the wizard. In Customization Script, add the content of the userdata file. The userdata file contains the IP address, Netmask and Gateway details, and customer scripts of the VPX instance.

7. After a key pair is selected or imported, check config-drive option and click **Launch**.


Launch Instance
×
Details *
Access & Security
Networking *
Post-Creation
Advanced Options
Cancel
Launch

Disk Partition: Automatic

☑ Configuration Drive

Specify advanced options to use when launching an instance.

Provision the VPX instance by using OpenStack CLI

Follow these steps to provision a VPX instance by using OpenStack CLI.

1. To create an image from qcow2, type the following command:

```
openstack image create --container-format bare --property hw_disk_bus=ide --diskformat  qcow2 --file NSVPX-OpenStack.qcow2 --public VPX-ToT-Image
```

2. To select an image for creating an instance, type the following command:

```
openstack image list | more
```

3. To create an instance of a particular flavor, type the following command to choose a flavor ID/Name of from a list:

```
openstack flavor list
```

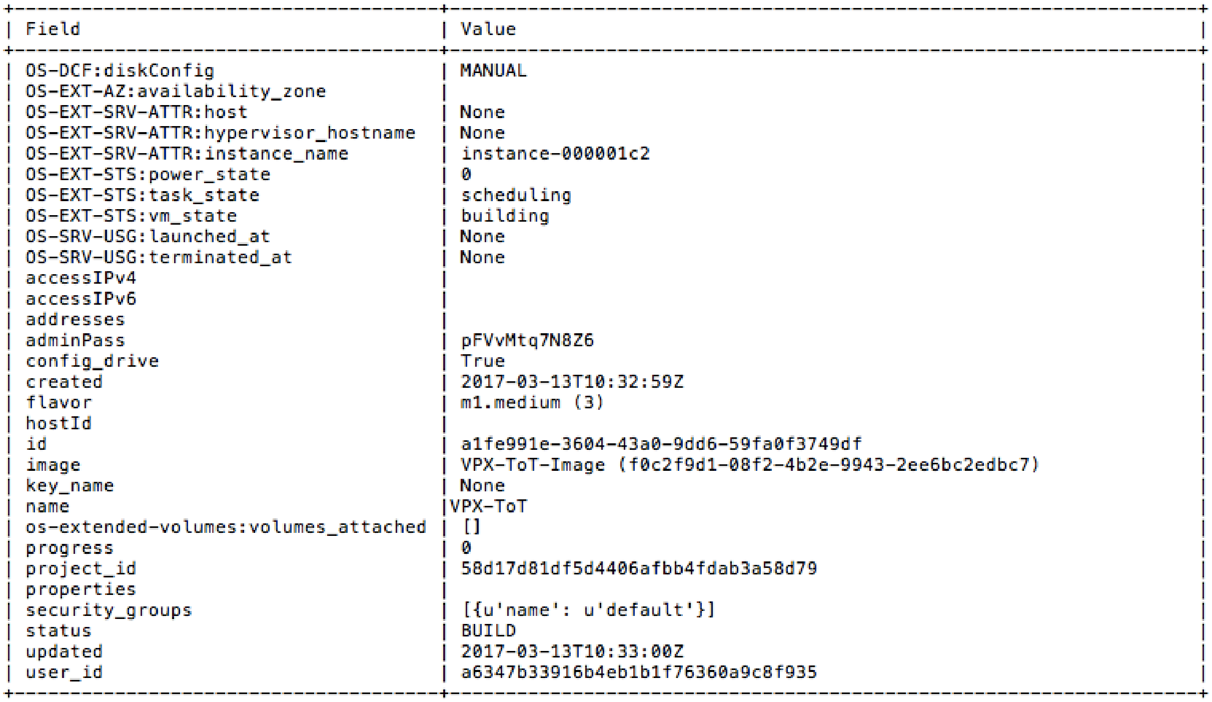
4. To attach a NIC to a particular network, type the following command to choose a network ID from a network list:

```
openstack network list
```

5. To create an instance, type the following command:

```
openstack server create --flavor FLAVOR_ID --image IMAGE_ID --key-name KEY_NAME
--user-data USER_DATA_FILE_PATH --config-drive True --nic net-id=net-uuid
INSTANCE_NAME
openstack server create --image VPX-ToT-Image --flavor m1.medium --user-data
ovf.xml --config-drive True --nic net-id=2734911b-ee2b-48d0-a1b6-3efd44b761b9
VPX-ToT
```

Figure 2: The following illustration provides a sample output.



```
+--------------------------------------+----------------------------------------------------------------------+
| Field                                | Value                                                                |
+--------------------------------------+----------------------------------------------------------------------+
| OS-DCF:diskConfig                    | MANUAL                                                               |
| OS-EXT-AZ:availability_zone          |                                                                      |
| OS-EXT-SRV-ATTR:host                 | None                                                                 |
| OS-EXT-SRV-ATTR:hypervisor_hostname  | None                                                                 |
| OS-EXT-SRV-ATTR:instance_name        | instance-000001c2                                                    |
| OS-EXT-STS:power_state               | 0                                                                    |
| OS-EXT-STS:task_state                | scheduling                                                           |
| OS-EXT-STS:vm_state                  | building                                                             |
| OS-SRV-USG:launched_at               | None                                                                 |
| OS-SRV-USG:terminated_at             | None                                                                 |
| accessIPv4                           |                                                                      |
| accessIPv6                           |                                                                      |
| addresses                            |                                                                      |
| adminPass                            | pFVvMtq7N8Z6                                                         |
| config_drive                         | True                                                                 |
| created                              | 2017-03-13T10:32:59Z                                                 |
| flavor                               | m1.medium (3)                                                        |
| hostId                               |                                                                      |
| id                                   | a1fe991e-3604-43a0-9dd6-59fa0f3749df                                 |
| image                                | VPX-ToT-Image (f0c2f9d1-08f2-4b2e-9943-2ee6bc2edbc7)                 |
| key_name                             | None                                                                 |
| name                                 |VPX-ToT                                                               |
| os-extended-volumes:volumes_attached |  []                                                                  |
| progress                             | 0                                                                    |
| project_id                           | 58d17d81df5d4406afbb4fdab3a58d79                                     |
| properties                           |                                                                      |
| security_groups                      |  [{u'name': u'default'}]                                             |
| status                               | BUILD                                                                |
| updated                              | 2017-03-13T10:33:00Z                                                 |
| user_id                              | a6347b33916b4eb1b1f76360a9c8f935                                     |
+--------------------------------------+----------------------------------------------------------------------+
```

Provisioning the NetScaler Virtual Appliance by using the Virtual Machine Manager

September 21, 2020

The Virtual Machine Manager is a desktop tool for managing VM Guests. It enables you to create new VM Guests and various types of storage, and manage virtual networks. You can access the graphical console of VM Guests with the built-in VNC viewer and view performance statistics, either locally or remotely.

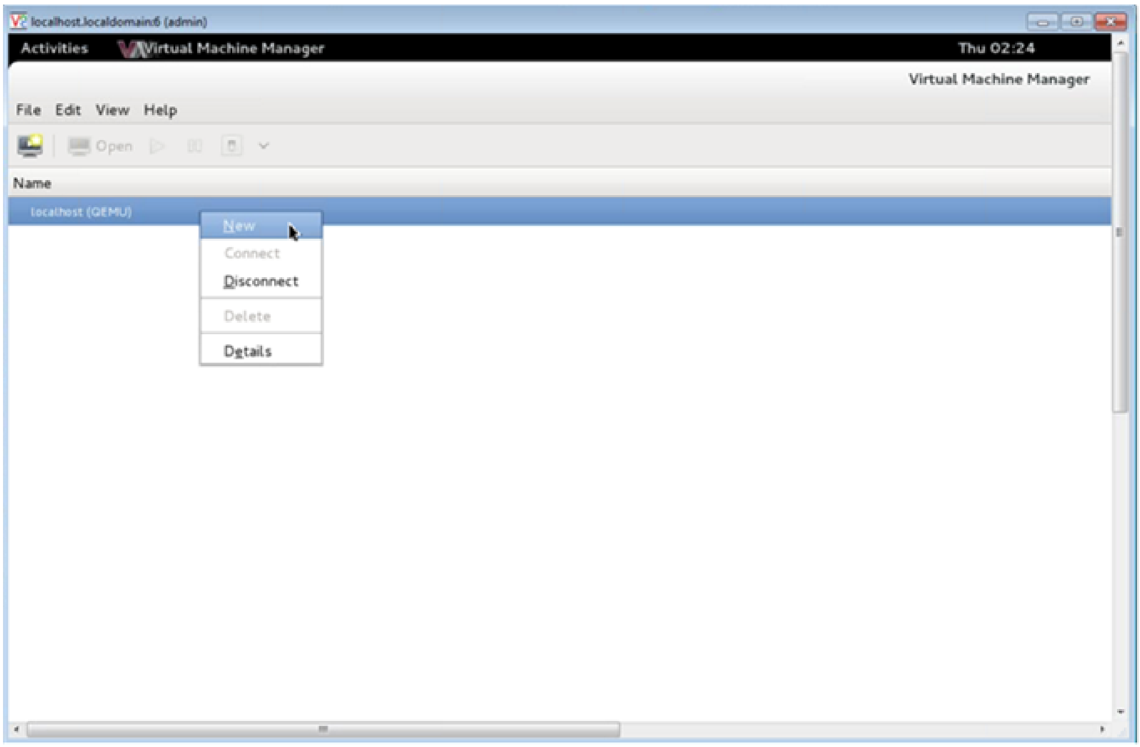
After installing your preferred Linux distribution, with KVM virtualization enabled, you can proceed with provisioning virtual machines.

Provisioning the NetScaler Virtual Appliance using the RAW Image

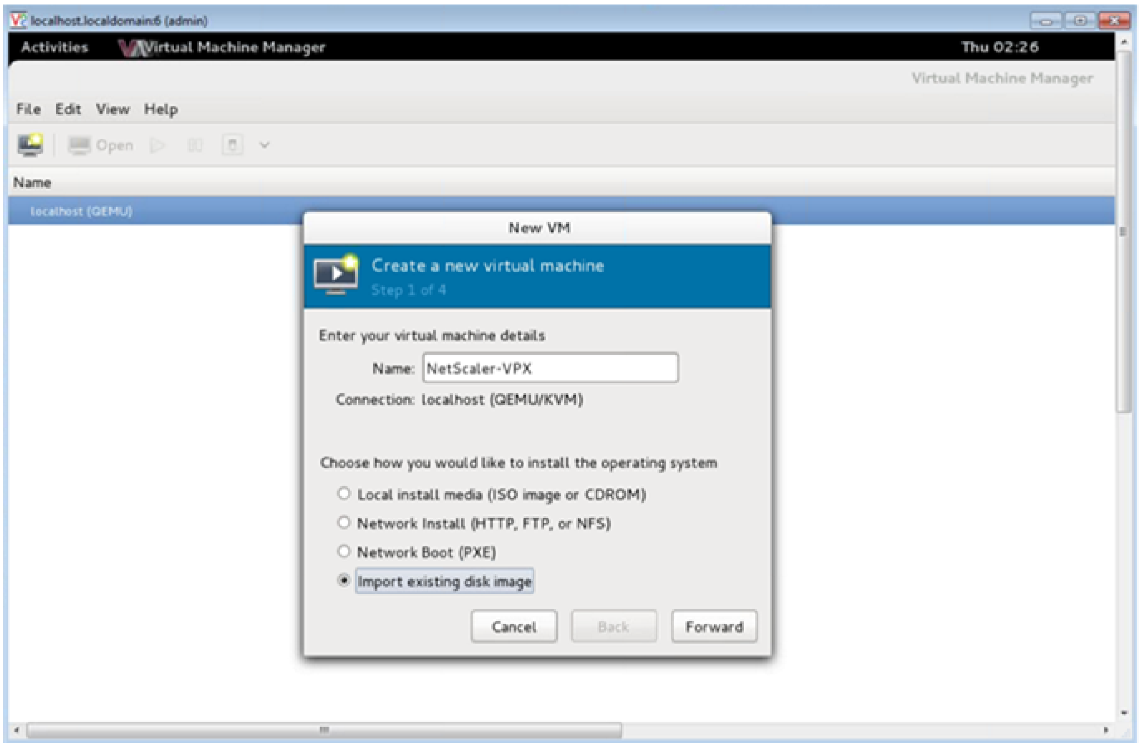
Using the Virtual Machine Manager, you can provision the NetScaler VPX using the RAW image.

To provision a NetScaler VPX by using Virtual Machine Manager, follow these steps.

1. Open the Virtual Machine Manager (**Application** > **System Tools** > **Virtual Machine Manager**) and enter the logon credentials in the **Authenticate** window.
2. Click the icon or right-click **localhost (QEMU)** to create a new NetScaler VPX instance.


localhost.localdomain6 (admin)
Activities
Virtual Machine Manager
Thu 02:24
Virtual Machine Manager
File Edit View Help
Open
Name
localhost (QEMU)
New
Connect
Disconnect
Delete
Details

3. In the **Name** text box, enter a name for the new VM (for example, NetScaler-VPX).
4. In the **New VM** window, under "Choose how you would like to install the operating system," select **Import existing disk image**, and then and click **Forward**.


localhost.localdomain6 (admin)
Activities
Virtual Machine Manager
Thu 02:26
Virtual Machine Manager
File Edit View Help
Open
Name
localhost (QEMU)
New VM

Create a new virtual machine
Step 1 of 4

Enter your virtual machine details

Name: NetScaler-VPX

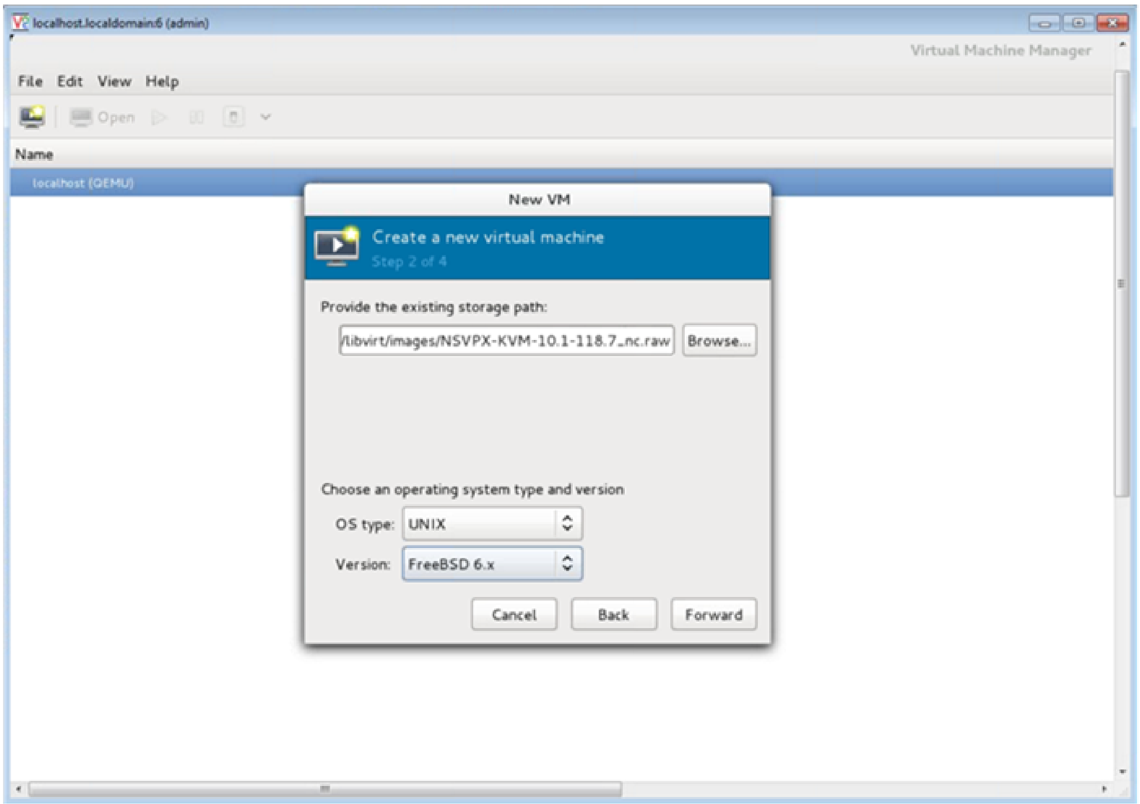
Connection: localhost (QEMU/KVM)

Choose how you would like to install the operating system

- ○ Local install media (ISO image or CDROM)
- ○ Network Install (HTTP, FTP, or NFS)
- ○ Network Boot (PXE)
- ● Import existing disk image

Cancel
Back
Forward

5. In the **Provide the existing storage path** field, navigate the path to the image. Choose the OS type as UNIX and Version as FreeBSD 6.x. Then, click **Forward**.



localhost.localdomain:6 (admin)
Virtual Machine Manager
File Edit View Help
Open
Name
localhost (QEMU)
New VM
Create a new virtual machine
Step 2 of 4
Browse...
Cancel
Back
Forward

Provide the existing storage path: /libvirt/images/NSVPX-KVM-10.1-118.7_nc.raw

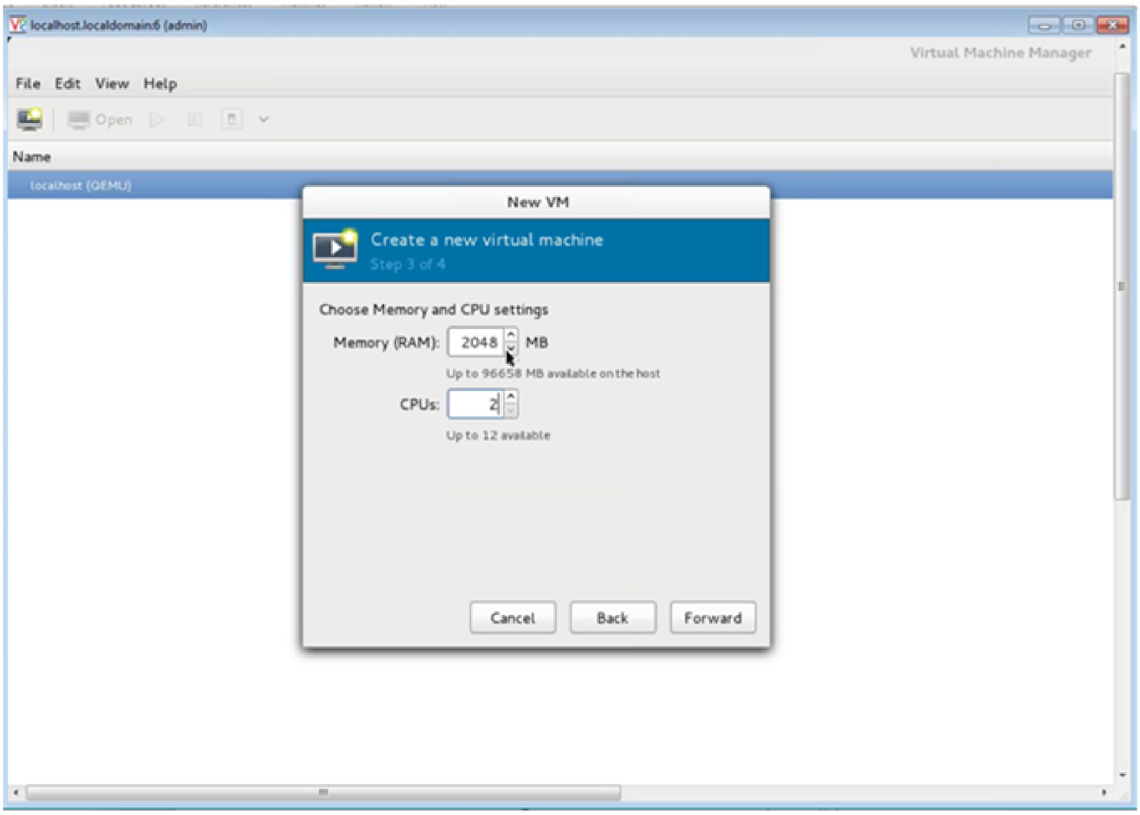
Choose an operating system type and version

OS type: UNIX

Version: FreeBSD 6.x

6. Under "Choose Memory and CPU settings," select the following settings, and then click **Forward**:

   - Memory (RAM)— 2048 MB
   - CPUs— 2


localhost.localdomain:6 (admin)
Virtual Machine Manager
File
Edit
View
Help
Open
Name
localhost (QEMU)
Cancel
Back
Forward

New VM

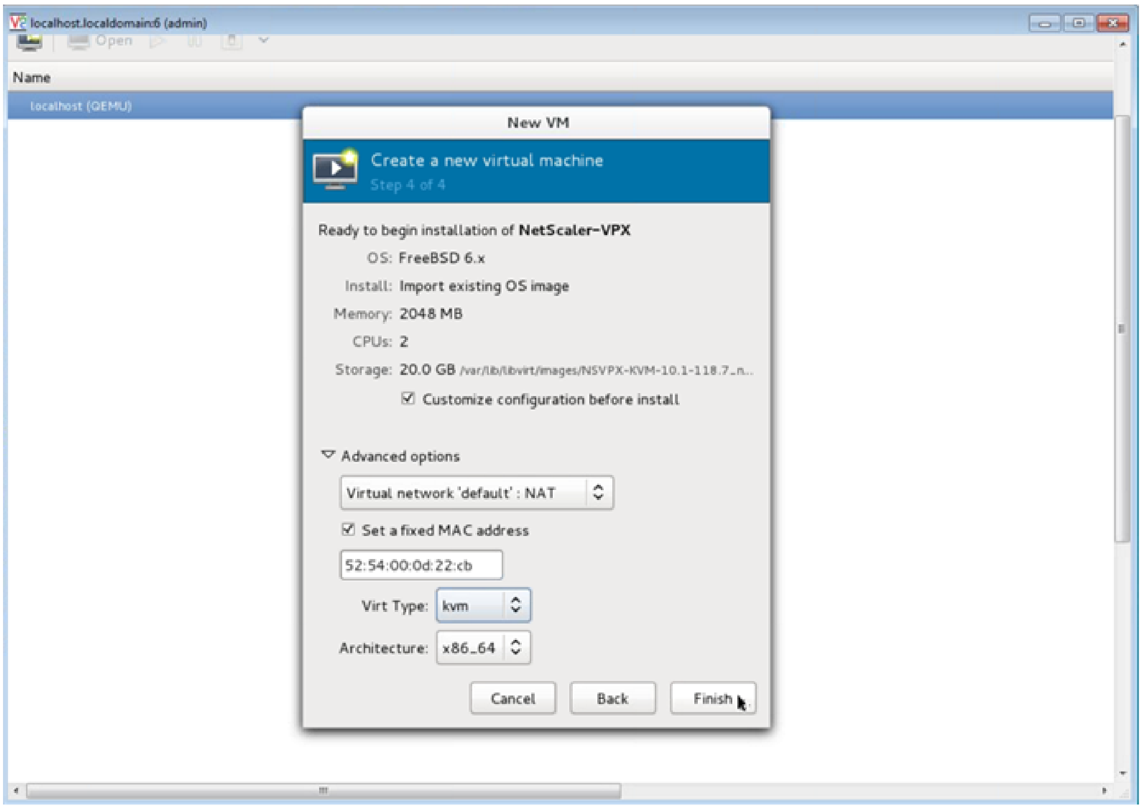
Create a new virtual machine
Step 3 of 4

Choose Memory and CPU settings

Memory (RAM): 2048 MB
Up to 96658 MB available on the host

CPUs: 2
Up to 12 available

7. Select the **Customize configuration before install** check box. Optionally, under "Advanced options," you can you can customize the MAC address. Make sure the **Virt Type** selected is kvm and the Architecture selected is x86_64. Click **Finish**.



localhost.localdomain:6 (admin)
Open
Name
localhost (QEMU)
New VM

Create a new virtual machine
Step 4 of 4

Ready to begin installation of **NetScaler-VPX**

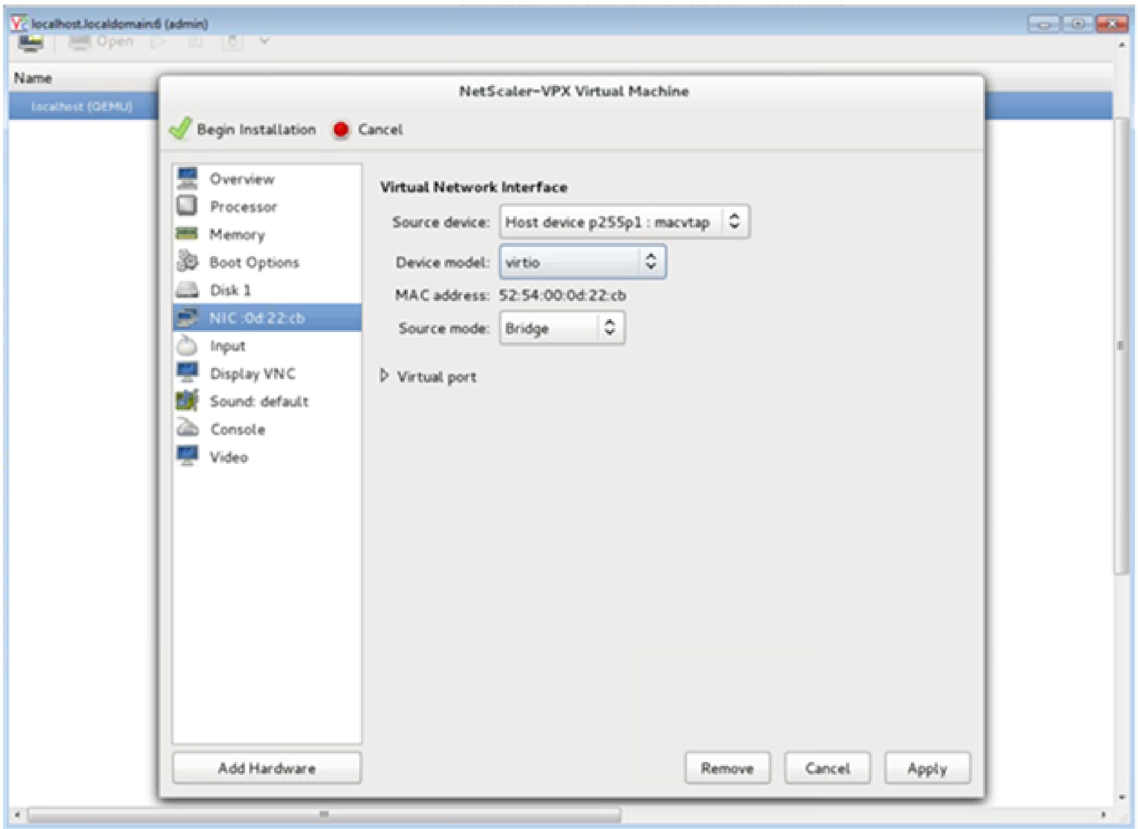
OS: FreeBSD 6.x
Install: Import existing OS image
Memory: 2048 MB
CPUs: 2
Storage: 20.0 GB /var/lib/libvirt/images/NSVPX-KVM-10.1-118.7_n...
☑ Customize configuration before install

Advanced options
Virtual network 'default' : NAT
☑ Set a fixed MAC address
52:54:00:0d:22:cb
Virt Type: kvm
Architecture: x86_64

Cancel
Back
Finish

8. Select a NIC and provide the following configuration:

   - Source device— ethX macvtap or Bridge
   - Device model— virtio
   - Source mode— Bridge


localhost.localdomain6 (admin)
Open
Name
localhost (QEMU)
NetScaler-VPX Virtual Machine
Begin Installation
Cancel
Overview
Processor
Memory
Boot Options
Disk 1
NIC :0d:22:cb
Input
Display VNC
Sound: default
Console
Video
Virtual port
Add Hardware
Remove
Cancel
Apply

Virtual Network Interface

Source device: Host device p255p1 : macvtap

Device model: virtio

MAC address: 52:54:00:0d:22:cb

Source mode: Bridge

9. Click **Apply**, and then click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces

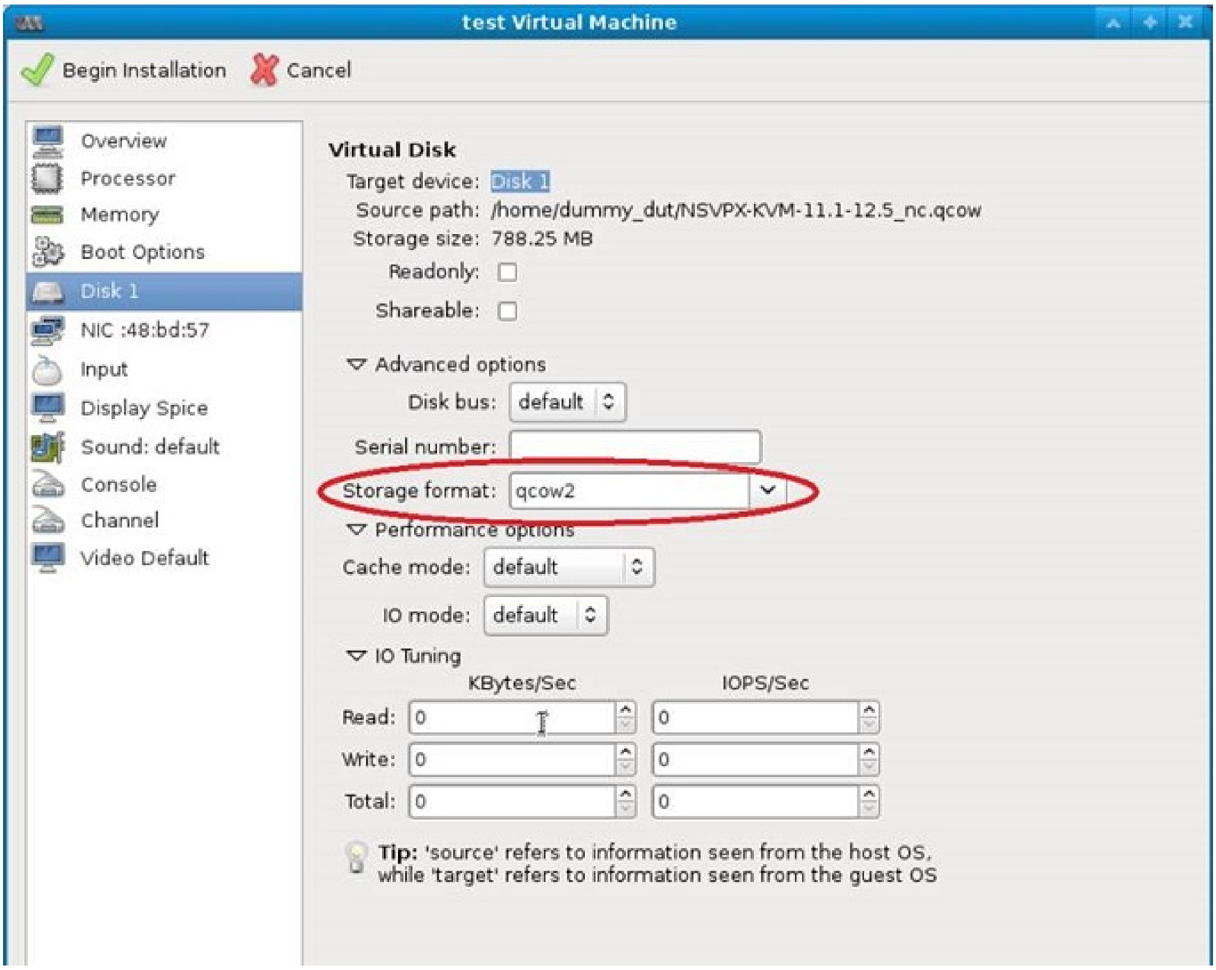
Provisioning the NetScaler Virtual Appliance by using the QCOW2 Image

Using the Virtual Machine Manager, you can provision the NetScaler VPX using the QCOW2 image.

Note: You can also convert the NetScaler VPX RAW image to QCOW2 image and provision the NetScaler VPX. For instructions to convert the RAW image to QCOW2, see Converting the RAW Image Format to a QCOW2 Image Format.

To provision the NetScaler VPX using QCOW2 image:

1. Follow **step 1** to **step 8** in Provisioning the NetScaler Virtual Appliance by using the RAW Image.
   Note: Make sure that you select **qcow2** image in **step 5**.
2. Select **Disk 1** and click **Advanced options**. Select **qcow2** from the Storage format drop-down list.


test Virtual Machine
Begin Installation
Cancel
Overview
Processor
Memory
Boot Options
Disk 1
NIC :48:bd:57
Input
Display Spice
Sound: default
Console
Channel
Video Default

Virtual Disk
Target device: Disk 1
Source path: /home/dummy_dut/NSVPX-KVM-11.1-12.5_nc.qcow
Storage size: 788.25 MB
Readonly: ☐
Shareable: ☐

Advanced options
Disk bus: default
Serial number:
Storage format: qcow2

Performance options
Cache mode: default
IO mode: default

IO Tuning

	KBytes/Sec	IOPS/Sec
Read:	0	0
Write:	0	0
Total:	0	0

Tip: 'source' refers to information seen from the host OS, while 'target' refers to information seen from the guest OS

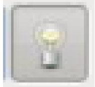
4. Click **Apply**, and then click **Begin Installation**. After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

Adding Additional Interfaces to NetScaler VPX by using Virtual Machine Manager

After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

To add additional interfaces, follow these steps:

1. Shut down the NetScaler VPX instance running on the KVM.
2. Right-click the VPX instance and choose **Open** from the pop-up menu.
3. Click the following icon in the header to view the virtual hardware details.



4/. Click **Add Hardware**. In the **Add New Virtual Hardware window**, select **Network** from the navigation menu

```
![image](/en-us/netscaler/media/hardware-network.png)
```

5/. In **Host Device** field, select the physical interface type. The host device type can be either Bridge or MacVTap. In case of MacVTap, four modes possible are VEPA, Bridge, Private and Pass-through.

```
1.  For Bridge
    1.  Host device – Select the "Specify shared device name" option.

    2.  Provide the Bridge name that is configured in the KVM host.

        Note: Make sure that you have configured a Linux bridge in the
            KVM host, bound the physical interface to the bridge, and
            put the bridge in the UP state.

        ![image](/en-us/netscaler/media/network-host-device.png)

    3.  Device model – virtio.

    4.  Click Finish.
2.  For MacVTap
    1.  Host device – Select the physical interface from the menu.

    2.  Device model – virtio.

        ![image](/en-us/netscaler/media/macvtap_image.jpg)

    3.  Click Finish. You can view the newly added NIC in the
       navigation pane.

        ![image](/en-us/netscaler/media/nic-finish.png)

    4.  Select the newly added NIC and select the Source mode for this
       NIC. The available modes are VEPA, Bridge, Private, and
       Passthrough. For more details on the interface and modes, see
       Source Interface and Modes.

    5.  Click Apply.
```

6/. Start the NetScaler VPX VM.

> **Important**
>
> Limition: Interface parameter configurations such as speed, duplex, and autonegotiation are not supported.

Converting the RAW Image Format to a QCOW2 Image Format

You can convert the NetScaler VPX RAW image to QCOW2 image and provision the NetScaler VPX. To convert the RAW image to QCOW2 image. At the command prompt, enter the following command:

qemu-img convert -O qcow2 original-image.raw image-converted.qcow

For example:
qemu-img convert -O qcow2 NSVPX-KVM-11.1-12.5_nc.raw NSVPX-KVM-11.1-12.5_nc.qcow

Configure a NetScaler VPX instance to use SR-IOV network interfaces

September 21, 2020

You can use the Virtual Machine Manager to configure a NetScaler VPX instance running on Linux-KVM to use single root I/O virtualization (SR-IOV) network interfaces with Intel 82599 10G NIC and X710 10G and XL710 40G NICs.

This section describes how to:

- Configure a NetScaler VPX Instance to Use SR-IOV Network Interface
- Configure Static LA/LACP on the SR-IOV Interface
- Configure VLAN on the SR-IOV Interface

Limitations

Keep the limitations in mind while using Intel 82599, X710, XL710 NICs. The following features not supported.

Limitations for Intel 82599 NICs:

- L2 mode switching
- Admin partitioning (shared VLAN mode)
- High availability (active-active mode
- Jumbo Frames.
- IPv6: You can configure only up to 30 unique IPv6 addresses in a VPX instance if you've alteast one SR-IOV interface.
- VLAN configuration on Hypervisor for SRIOV VF interface through "ip link" command is not supported.
- Interface parameter configurations such as speed, duplex, and autonegotiations are not supported.

Limitations for X710 10G and XL710 40G NICs:

- L2 mode switching.
- In a cluster, Jumbo Frames are not supported when the XL710 NIC is used as a data interface.
- Interface list re-orders when interfaces are disconnected and reconnected.
- Interface parameter configurations such as speed, duplex, and auto negotiations are not supported.
- Interface name is 40/X for both XL710 and X710 NICs
- Up to 16 Intel XL710/X710 SRIOV or PCI Passthrough interfaces can be supported on a VPX instance.

> Note: For IPv6 to work with X710 10G and XL710 40G NICs, you need to enable trust mode on the Virtual Functions (VFs) by typing the following command on the KVM host:

```
## ip link set <PNIC> <VF> trust on
```

For example:

```
## ip link set ens785f1 vf 0 trust on
```

Prerequisites

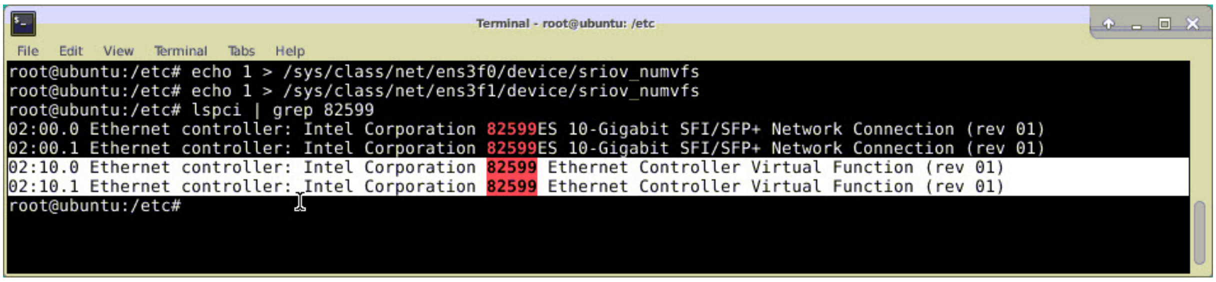
Before you configure a NetScaler VPX instance to use SR-IOV network interfaces, complete the following prerequisite tasks. See the NIC column for details about about how to complete the corresponding tasks.

Task	82599 NIC	X710 and XL710 NICs
1. Add the NIC to the KVM host.	-	-
2. Download and install the latest Intel driver.	IXGBE driver	I40E driver
3. Blacklist the driver on the KVM host.	Add the following entry in the /etc/modprobe.d/blacklist.conf file: *blacklist ixgbevf*. Use IXGBE driver version 4.3.15 (recommended).	Add the following entry in the /etc/modprobe.d/blacklist.conf file: *blacklist i40evf*.Use i40e driver version 2.0.26 (recommended).

Task	82599 NIC	X710 and XL710 NICs
4.Enable SR-IOV Virtual Functions (VFs) on the KVM host. In both the commands in the next two columns: *number_of_VFs* =the number of Virtual VFs that you want to create. *device_name* =the interface name.	If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/ixgbe file and restart the KVM host: *options ixgbe max_vfs=*.If you are using kernel 3.8 version or later, create VFs using the following command: *echo > /sys/class/net//device/sriov_numvfs*. See example in figure 1.	If you are using earlier version of kernel 3.8, then add the following entry to the /etc/modprobe.d/i40e.conf file and restart the KVM host:*options i40e max_vfs=*.If you are using kernel 3.8 version or later, create VFs using the following command: *echo > /sys/class/net//device/sriov_numvfs*.See example in figure 2.
5. Make the VFs persistent by adding the commands that you used to create VFs, to the rc.local file.	See example in figure 3.	See example in figure 3.

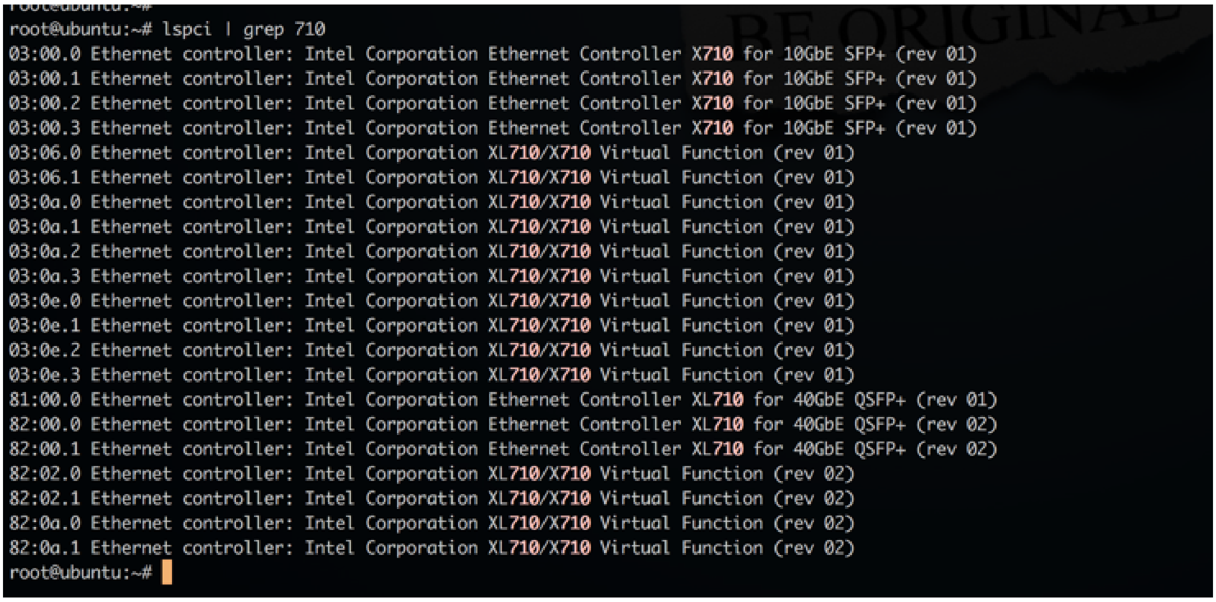
> **Important**
>
> When you are create the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

Figure 1: Enable SR-IOV VFs on the KVM host for 82599 10G NIC.



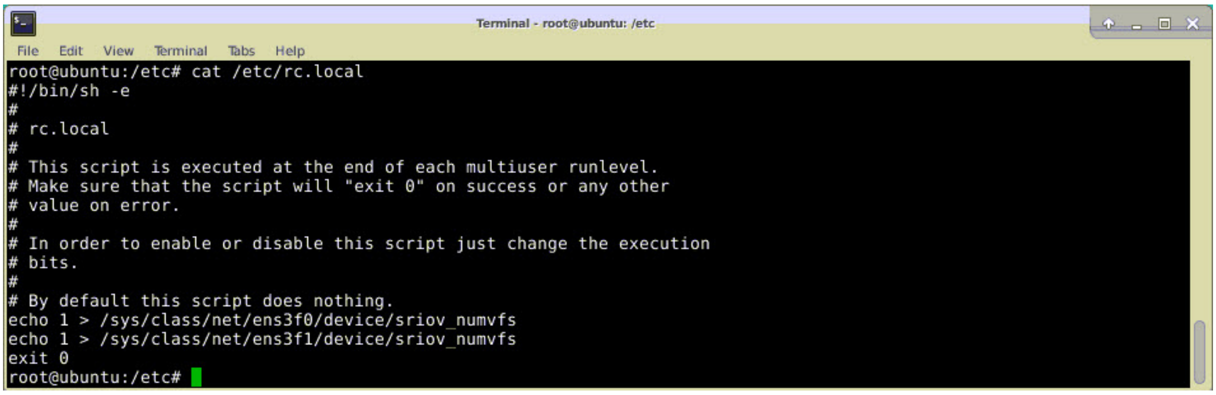
```
root@ubuntu:/etc# echo 1 > /sys/class/net/ens3f0/device/sriov_numvfs
root@ubuntu:/etc# echo 1 > /sys/class/net/ens3f1/device/sriov_numvfs
root@ubuntu:/etc# lspci | grep 82599
02:00.0 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
02:00.1 Ethernet controller: Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Connection (rev 01)
02:10.0 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
02:10.1 Ethernet controller: Intel Corporation 82599 Ethernet Controller Virtual Function (rev 01)
root@ubuntu:/etc#
```

Figure 2: Enable SR-IOV VFs on the KVM host for X710 10G and XL710 40G NICs.



```
root@ubuntu:~# lspci | grep 710
03:00.0 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.1 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.2 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:00.3 Ethernet controller: Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (rev 01)
03:06.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:06.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.2 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0a.3 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.2 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
03:0e.3 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 01)
81:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 01)
82:00.0 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 02)
82:00.1 Ethernet controller: Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (rev 02)
82:02.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:02.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:0a.0 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
82:0a.1 Ethernet controller: Intel Corporation XL710/X710 Virtual Function (rev 02)
root@ubuntu:~#
```

Figure 3: Make the VFs persistent.



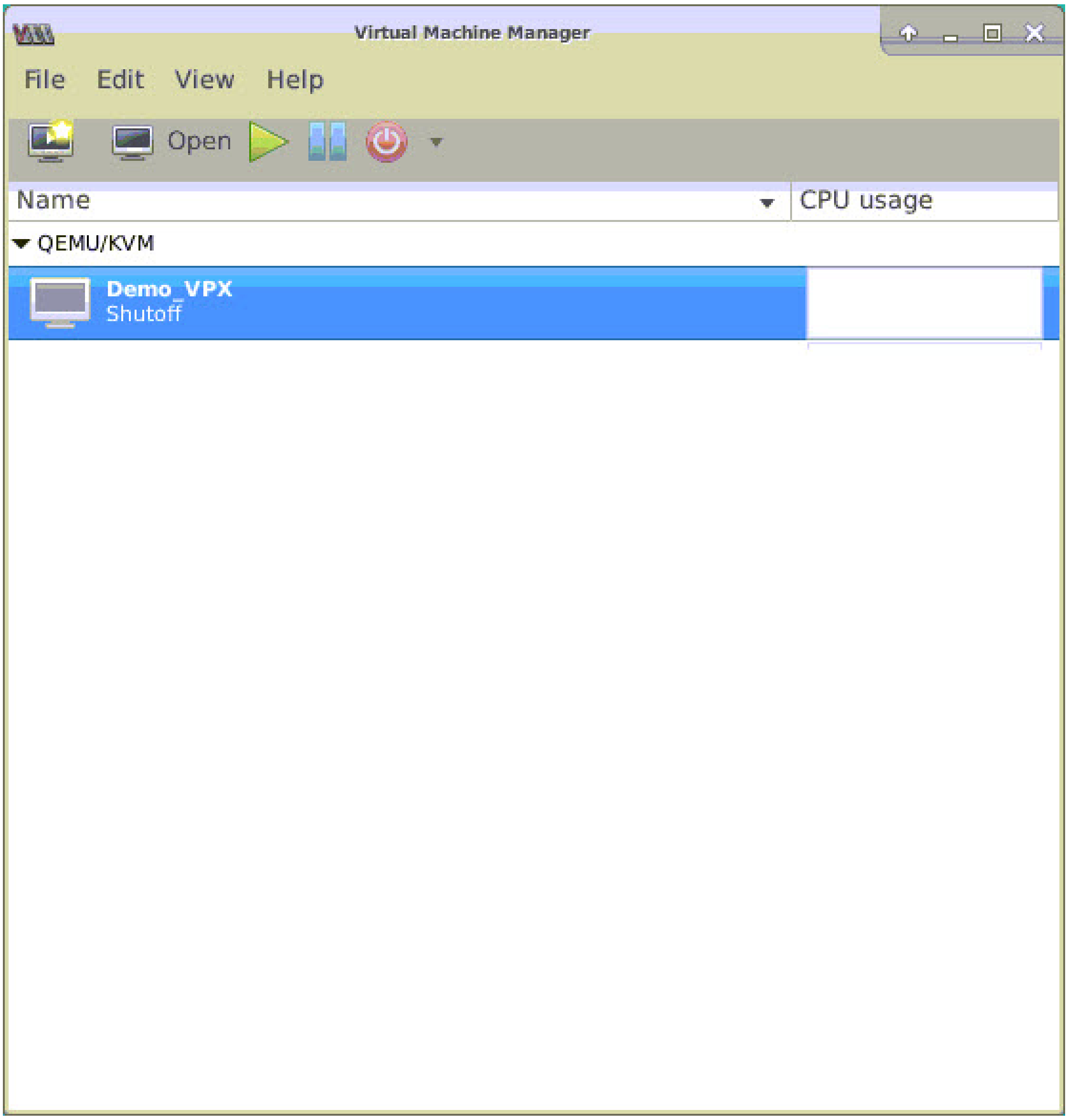
```
root@ubuntu:/etc# cat /etc/rc.local
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.
echo 1 > /sys/class/net/ens3f0/device/sriov_numvfs
echo 1 > /sys/class/net/ens3f1/device/sriov_numvfs
exit 0
root@ubuntu:/etc#
```

Configure a NetScaler VPX instance to use SR-IOV network interface

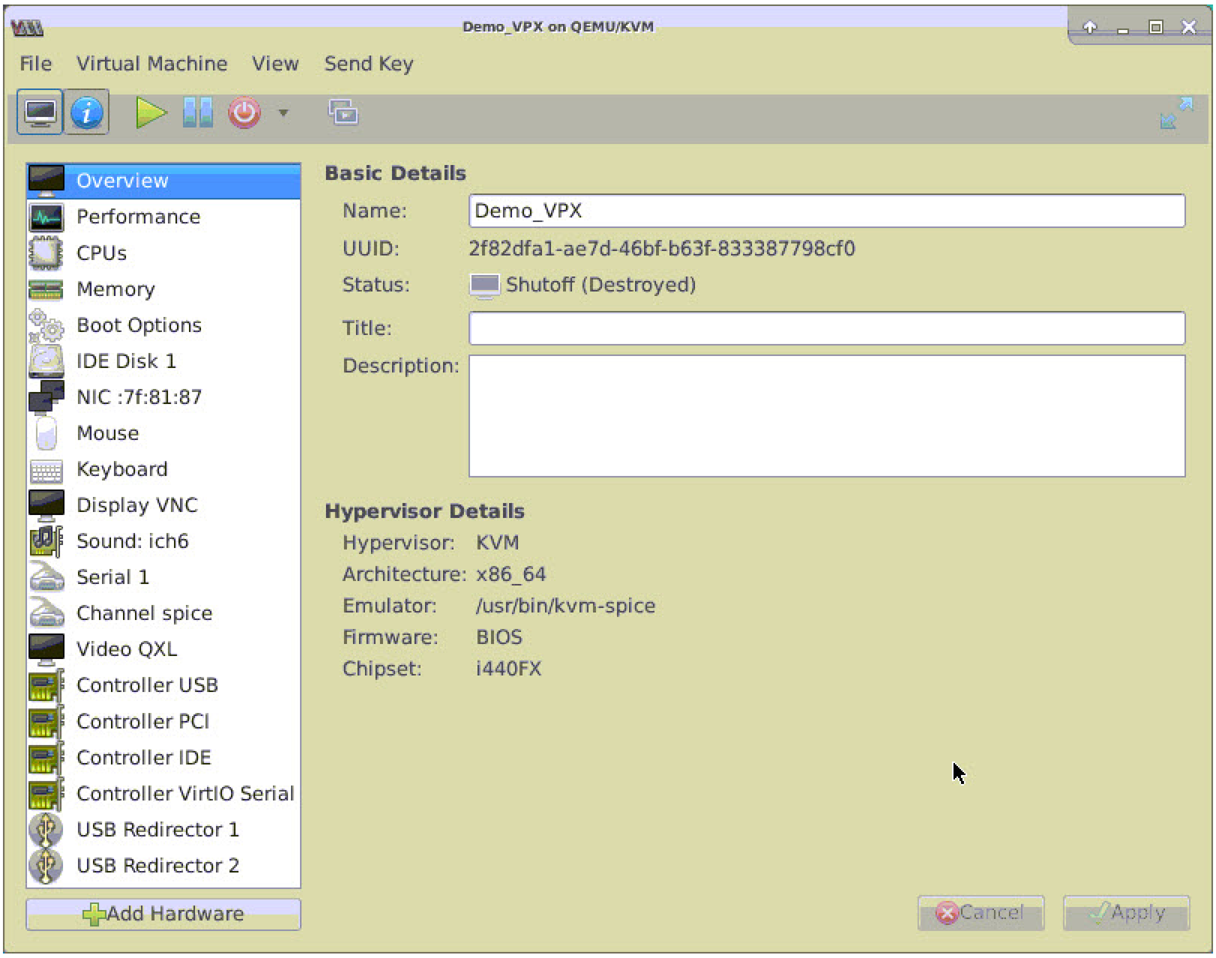
To configure NetScaler VPX instance to use SR-IOV network interface by using Virtual Machine Manager, complete these steps:

1. Power off the NetScaler VPX instance.

2. Select the NetScaler VPX instance and then select Open.


Virtual Machine Manager
File
Edit
View
Help
Open
Name
CPU usage
QEMU/KVM
Demo_VPX
Shutoff

1. In the <virtual_machine on KVM> window, select the **i** icon.


Demo_VPX on QEMU/KVM
File
Virtual Machine
View
Send Key
Overview
Performance
CPUs
Memory
Boot Options
IDE Disk 1
NIC :7f:81:87
Mouse
Keyboard
Display VNC
Sound: ich6
Serial 1
Channel spice
Video QXL
Controller USB
Controller PCI
Controller IDE
Controller VirtIO Serial
USB Redirector 1
USB Redirector 2
Add Hardware
Cancel
Apply

Basic Details

Name: Demo_VPX

UUID: 2f82dfa1-ae7d-46bf-b63f-833387798cf0

Status: Shutoff (Destroyed)

Title:

Description:

Hypervisor Details

Hypervisor: KVM

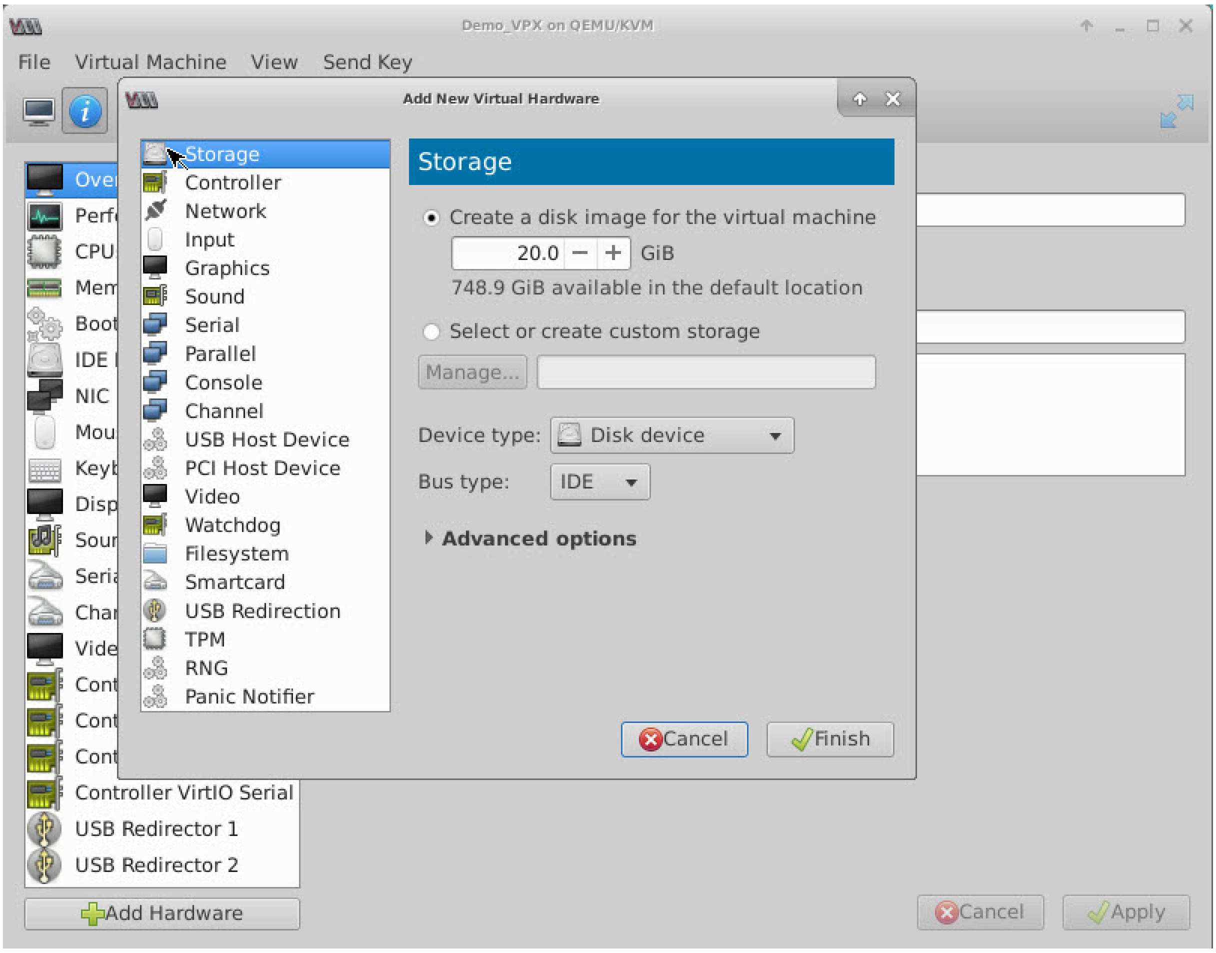
Architecture: x86_64

Emulator: /usr/bin/kvm-spice

Firmware: BIOS

Chipset: i440FX

1. Select **Add Hardware**.



Demo_VPX on QEMU/KVM
File
Virtual Machine
View
Send Key
Overview
Performance
CPUs
Memory
Boot Options
IDE Disk
NIC
Mouse
Keyboard
Display
Sound
Serial
Channel
Video
Controller
Controller
Controller
Controller VirtIO Serial
USB Redirector 1
USB Redirector 2
Add Hardware
Cancel
Apply
Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
Storage
Manage...
Advanced options
Cancel
Finish

● Create a disk image for the virtual machine

20.0 GiB

748.9 GiB available in the default location

○ Select or create custom storage

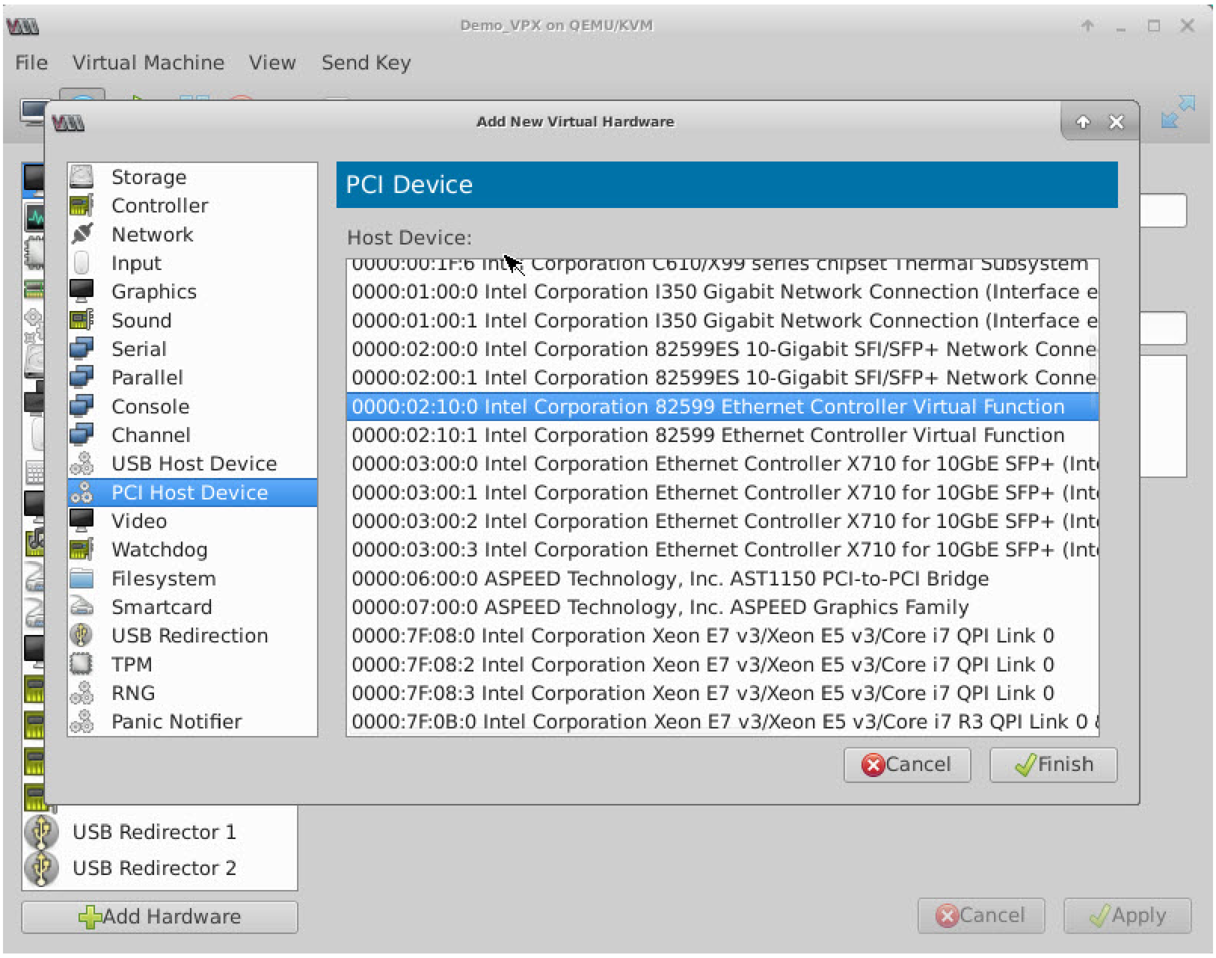
Device type: Disk device

Bus type: IDE

5. In the Add New Virtual Hardware dialog box, do the following:

```
a. Select PCI Host Device.
b. In the Host Device section, select the VF you have created and click
   Finish.
```

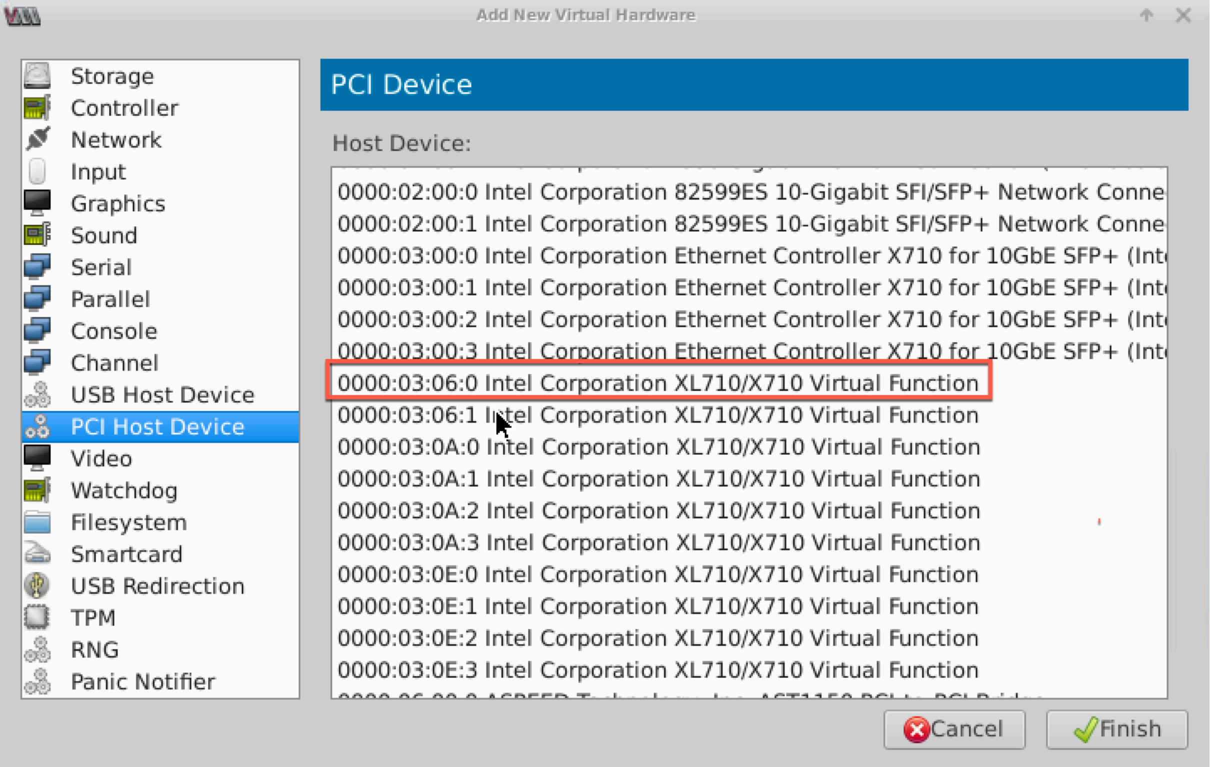
Figure 4:VF for 82599 10G NIC


Demo_VPX on QEMU/KVM
File
Virtual Machine
View
Send Key
Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
PCI Device
Cancel
Finish
USB Redirector 1
USB Redirector 2
Add Hardware
Cancel
Apply

Host Device:

0000:00:1F:6 Intel Corporation C610/X99 series chipset Thermal Subsystem
0000:01:00:0 Intel Corporation I350 Gigabit Network Connection (Interface e
0000:01:00:1 Intel Corporation I350 Gigabit Network Connection (Interface e
0000:02:00:0 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
0000:02:00:1 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
0000:02:10:0 Intel Corporation 82599 Ethernet Controller Virtual Function
0000:02:10:1 Intel Corporation 82599 Ethernet Controller Virtual Function
0000:03:00:0 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:1 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:2 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:3 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:06:00:0 ASPEED Technology, Inc. AST1150 PCI-to-PCI Bridge
0000:07:00:0 ASPEED Technology, Inc. ASPEED Graphics Family
0000:7F:08:0 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:08:2 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:08:3 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:0B:0 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 R3 QPI Link 0

Figure 5: VF for XL710 NIC


Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
PCI Device
Host Device:
Cancel
Finish

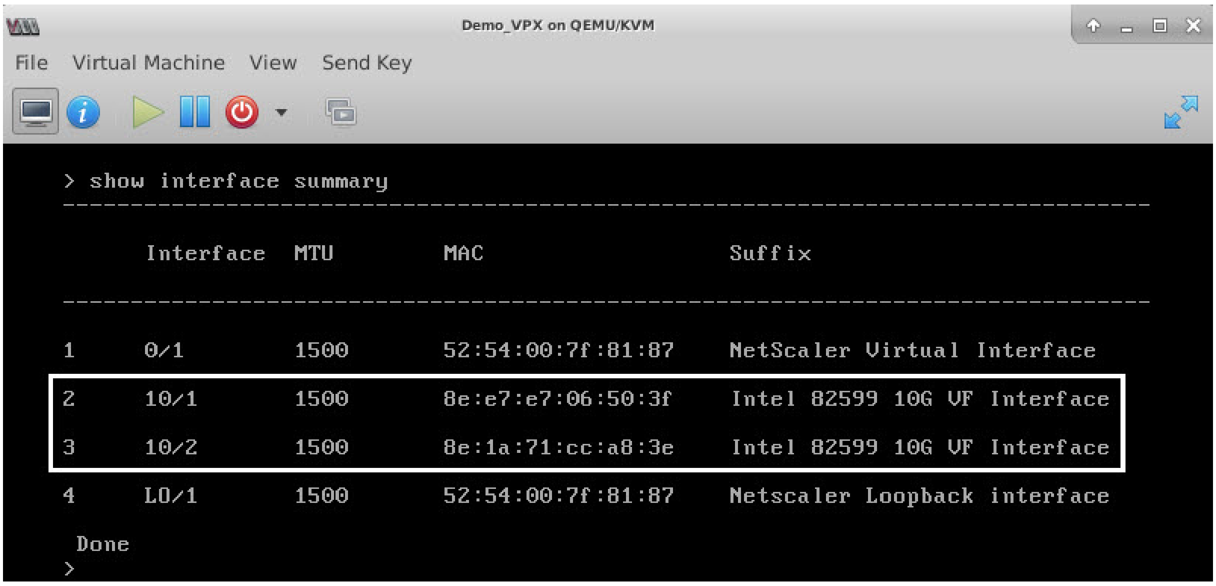
0000:02:00:0 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
0000:02:00:1 Intel Corporation 82599ES 10-Gigabit SFI/SFP+ Network Conne
0000:03:00:0 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:1 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:2 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:00:3 Intel Corporation Ethernet Controller X710 for 10GbE SFP+ (Int
0000:03:06:0 Intel Corporation XL710/X710 Virtual Function
0000:03:06:1 Intel Corporation XL710/X710 Virtual Function
0000:03:0A:0 Intel Corporation XL710/X710 Virtual Function
0000:03:0A:1 Intel Corporation XL710/X710 Virtual Function
0000:03:0A:2 Intel Corporation XL710/X710 Virtual Function
0000:03:0A:3 Intel Corporation XL710/X710 Virtual Function
0000:03:0E:0 Intel Corporation XL710/X710 Virtual Function
0000:03:0E:1 Intel Corporation XL710/X710 Virtual Function
0000:03:0E:2 Intel Corporation XL710/X710 Virtual Function
0000:03:0E:3 Intel Corporation XL710/X710 Virtual Function

6. Repeat Step 4 and 5 to add the VFs that you have created.
7. Power on the NetScaler VPX instance.
8. After the NetScaler VPX instance powers on, use the following command to verify the configuration:

```
> show interface summary
```

The output shows all the interfaces that you configured.

Figure 6: output summary for 82599.



```
> show interface summary
--------------------------------------------------------------------------------

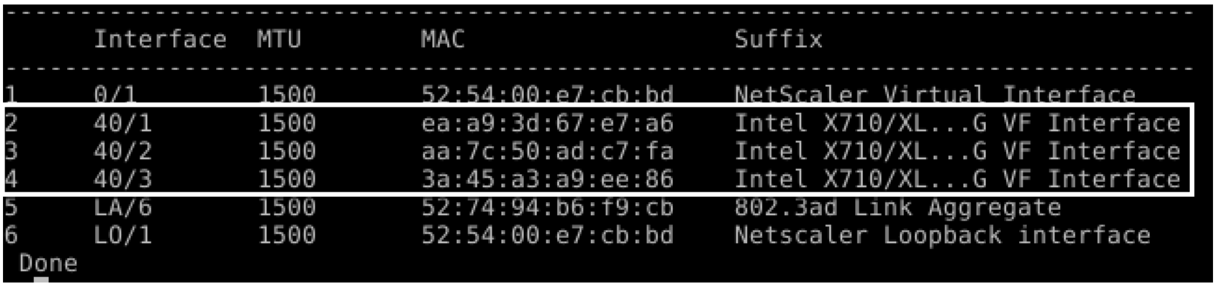
      Interface   MTU        MAC                   Suffix

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

1     0/1         1500       52:54:00:7f:81:87     NetScaler Virtual Interface
2     10/1        1500       8e:e7:e7:06:50:3f     Intel 82599 10G VF Interface
3     10/2        1500       8e:1a:71:cc:a8:3e     Intel 82599 10G VF Interface
4     LO/1        1500       52:54:00:7f:81:87     Netscaler Loopback interface

 Done
>
```

Figure 7. Output summary for X710 and XL710 NICs.



```
---------------------------------------------------------------------------------
      Interface  MTU         MAC                    Suffix
---------------------------------------------------------------------------------
1     0/1        1500        52:54:00:e7:cb:bd      NetScaler Virtual Interface
2     40/1       1500        ea:a9:3d:67:e7:a6      Intel X710/XL...G VF Interface
3     40/2       1500        aa:7c:50:ad:c7:fa      Intel X710/XL...G VF Interface
4     40/3       1500        3a:45:a3:a9:ee:86      Intel X710/XL...G VF Interface
5     LA/6       1500        52:74:94:b6:f9:cb      802.3ad Link Aggregate
6     LO/1       1500        52:54:00:e7:cb:bd      Netscaler Loopback interface
 Done
```

Configure static LA/LACP on the SR-IOV interface

> **Important**
>
> When you are creating the SR-IOV VFs, ensure that you do not assign MAC addresses to the VFs.

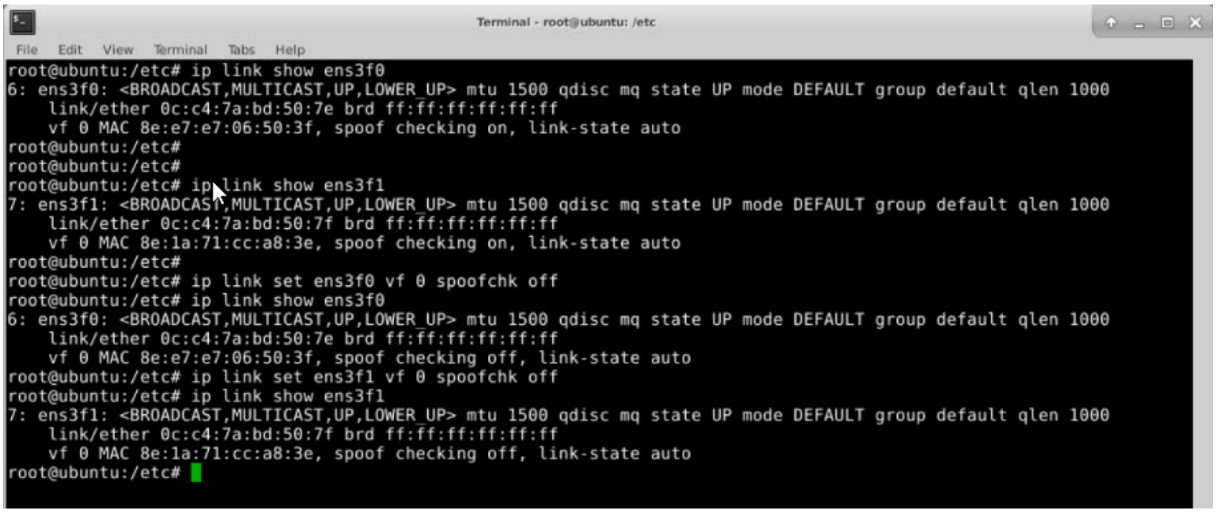
To use the SR-IOV VFs in link aggregation mode, disable spoof checking for VFs that you have created. On the KVM host, use the following command to disable spoof checking:

`*ip link set \<interface\\_name\> vf \<VF\\_id\> spoofchk off*`

Where:

- Interface_name – is the interface name.
- VF_id – is the Virtual Function id.

For example:



```
root@ubuntu:/etc# ip link show ens3f0
6: ens3f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 0c:c4:7a:bd:50:7e brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 8e:e7:e7:06:50:3f, spoof checking on, link-state auto
root@ubuntu:/etc#
root@ubuntu:/etc#
root@ubuntu:/etc# ip link show ens3f1
7: ens3f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 0c:c4:7a:bd:50:7f brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 8e:1a:71:cc:a8:3e, spoof checking on, link-state auto
root@ubuntu:/etc#
root@ubuntu:/etc# ip link set ens3f0 vf 0 spoofchk off
root@ubuntu:/etc# ip link show ens3f0
6: ens3f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 0c:c4:7a:bd:50:7e brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 8e:e7:e7:06:50:3f, spoof checking off, link-state auto
root@ubuntu:/etc# ip link set ens3f1 vf 0 spoofchk off
root@ubuntu:/etc# ip link show ens3f1
7: ens3f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode DEFAULT group default qlen 1000
    link/ether 0c:c4:7a:bd:50:7f brd ff:ff:ff:ff:ff:ff
    vf 0 MAC 8e:1a:71:cc:a8:3e, spoof checking off, link-state auto
root@ubuntu:/etc#
```

After you disable spoof checking for all the VFs that you have created. Restart the NetScaler VPX instance and configure link aggregation. For detailed instructions, see Configure Link Aggregation.

Configuring VLAN on the SR-IOV Interface

You can configure VLAN on SR-IOV VFs. For detailed instructions, see Configuring a VLAN.

> **Important**
>
> Ensure that the KVM host does not contain VLAN settings for the VF interface.

Configure a NetScaler VPX instance to use PCI passthrough network interfaces

September 21, 2020

After you have installed and configured a NetScaler VPX instance on the Linux-KVM platform, you can use the Virtual Machine Manager to configure the virtual appliance to use PCI passthrough network interfaces.

Prerequisites

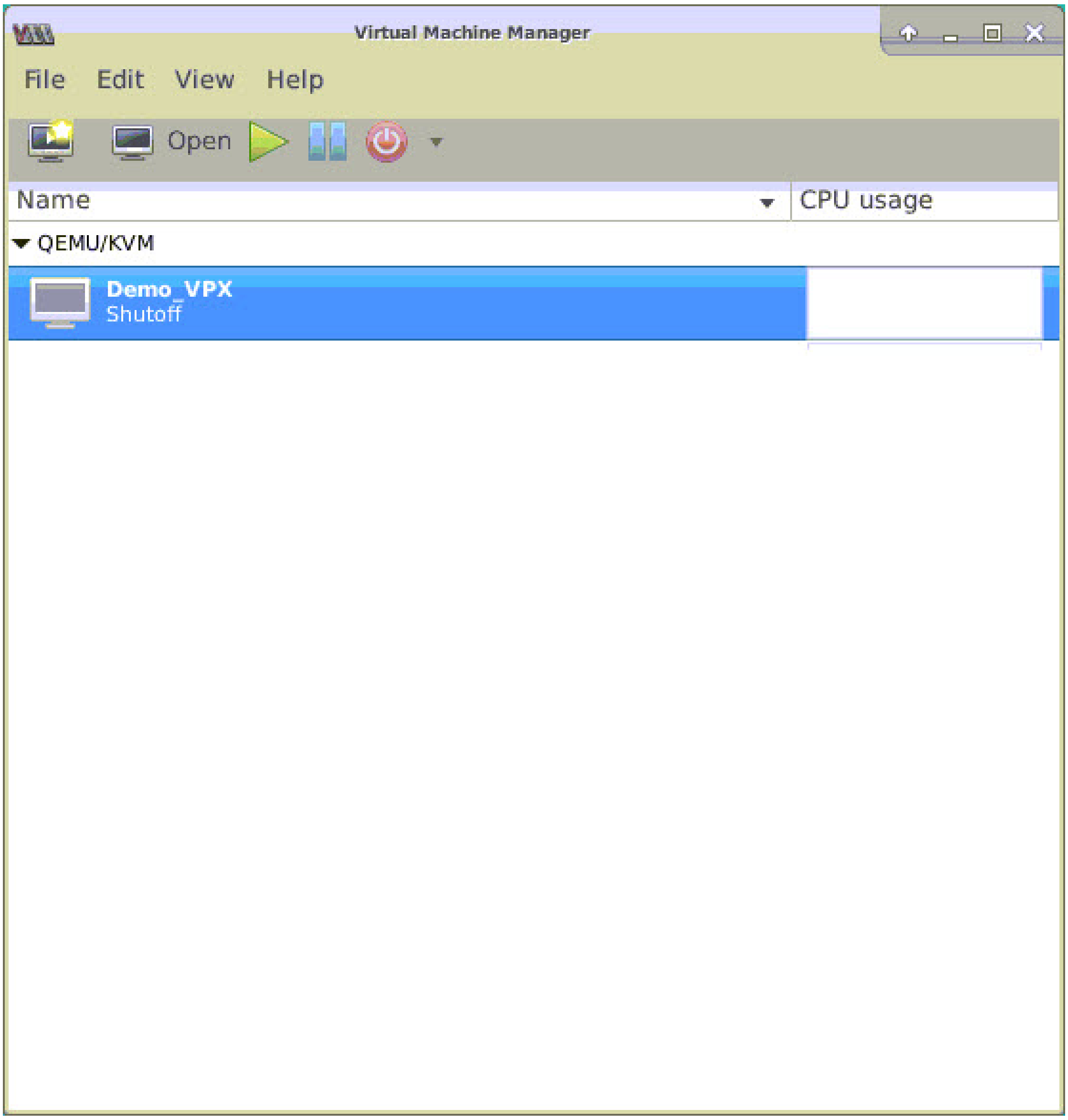
- The firmware version of the Intel XL710 Network Interface Card (NIC) on the KVM Host is 5.04.
- The KVM Host supports input–output memory management unit (IOMMU) and Intel VT-d, and they are enabled in the BIOS of the KVM Host. On the KVM Host, to enable IOMMU, add the following entry to the **/boot/grub2/grub.cfg** file:
  intel_iommu=1

- Execute the following command and reboot the KVM Host:
  Grub2-mkconfig –o /boot/grub2/grub.cfg

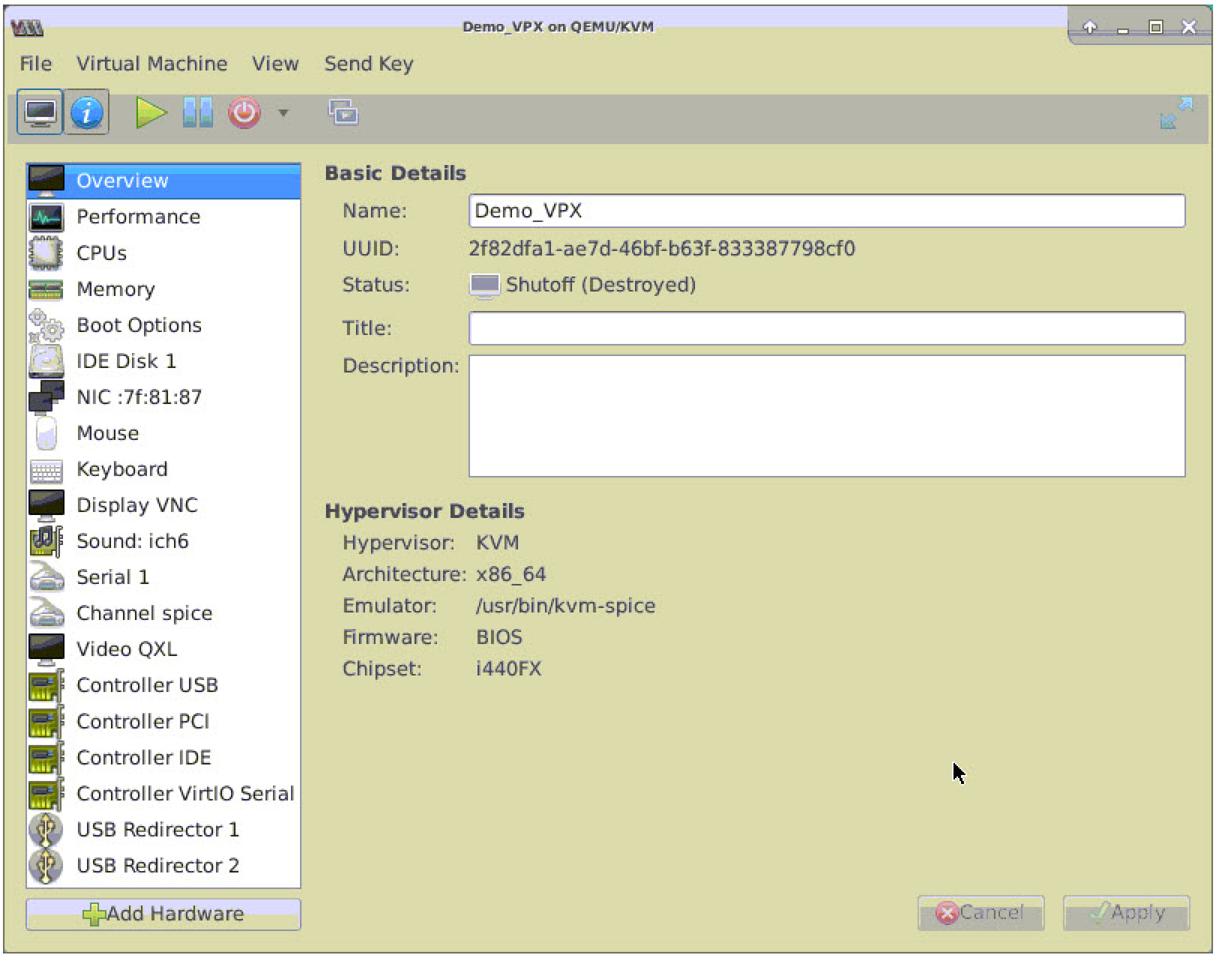
To configure NetScaler VPX instances to use PCI passthrough network interfaces by using the Virtual Machine Manager:

1. Power off the NetScaler VPX instance.

2. Select the NetScaler VPX instance and click **Open**.


Virtual Machine Manager
File
Edit
View
Help
Open
Name
CPU usage
QEMU/KVM
Demo_VPX
Shutoff

3. In the ***<virtual_machine on KVM>*** window, click the **i** icon.


Demo_VPX on QEMU/KVM
File
Virtual Machine
View
Send Key
Overview
Performance
CPUs
Memory
Boot Options
IDE Disk 1
NIC :7f:81:87
Mouse
Keyboard
Display VNC
Sound: ich6
Serial 1
Channel spice
Video QXL
Controller USB
Controller PCI
Controller IDE
Controller VirtIO Serial
USB Redirector 1
USB Redirector 2
Add Hardware
Cancel
Apply

Basic Details

Name: Demo_VPX
UUID: 2f82dfa1-ae7d-46bf-b63f-833387798cf0
Status: Shutoff (Destroyed)
Title:
Description:

Hypervisor Details

Hypervisor: KVM
Architecture: x86_64
Emulator: /usr/bin/kvm-spice
Firmware: BIOS
Chipset: i440FX

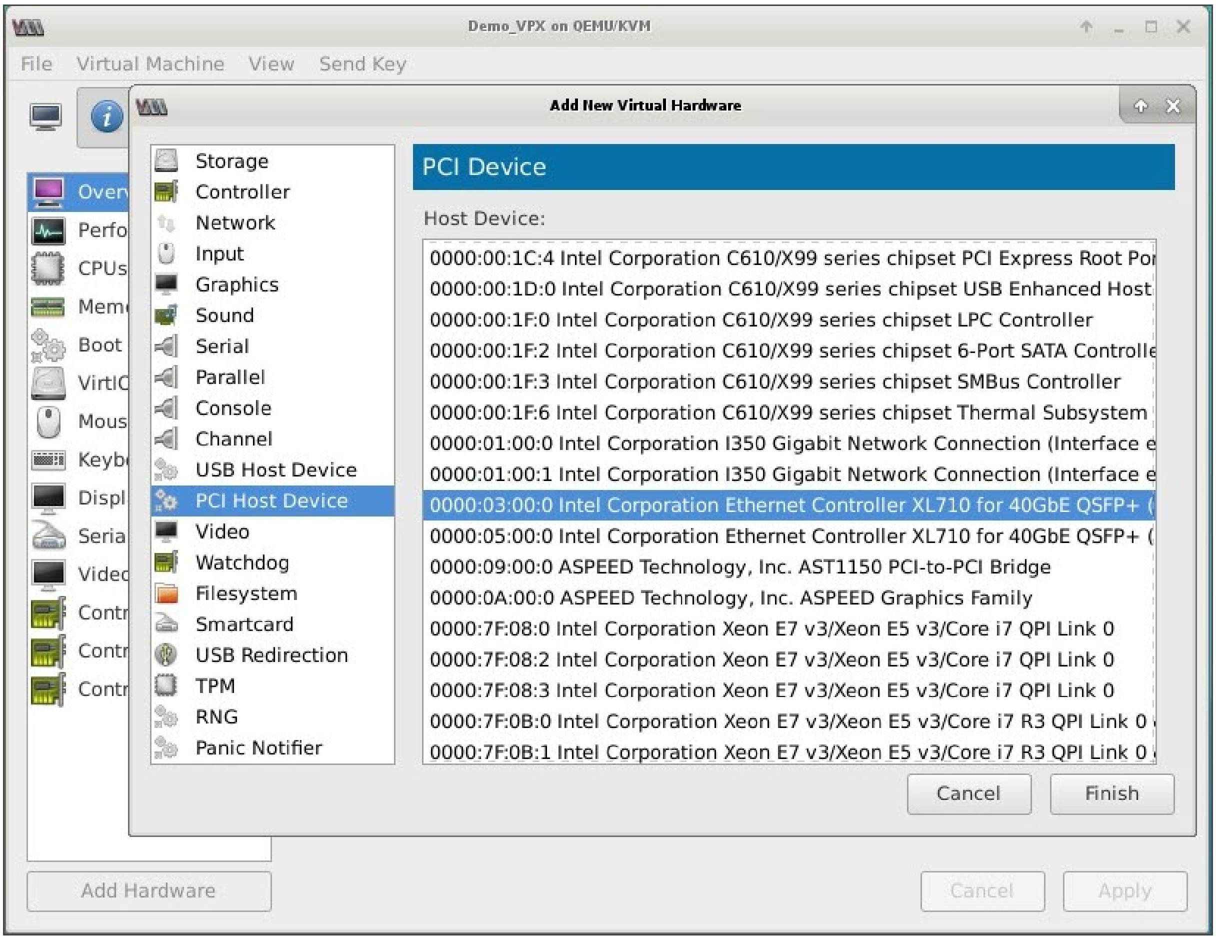
4. Click **Add Hardware**.

5. In the **Add New Virtual Hardware** dialog box, do the following:

   a. Select **PCI Host Device**.

   b. In the **Host Device** section, select the Intel XL710 physical function.

   c. Click **Finish**.


Demo_VPX on QEMU/KVM
File
Virtual Machine
View
Send Key
Overv
Perfo
CPUs
Mem
Boot
VirtIC
Mous
Keyb
Displ
Seria
Videc
Contr
Contr
Contr
Add Hardware
Add New Virtual Hardware
Storage
Controller
Network
Input
Graphics
Sound
Serial
Parallel
Console
Channel
USB Host Device
PCI Host Device
Video
Watchdog
Filesystem
Smartcard
USB Redirection
TPM
RNG
Panic Notifier
PCI Device
Host Device:
Cancel
Finish
Cancel
Apply

```
0000:00:1C:4 Intel Corporation C610/X99 series chipset PCI Express Root Po
0000:00:1D:0 Intel Corporation C610/X99 series chipset USB Enhanced Host
0000:00:1F:0 Intel Corporation C610/X99 series chipset LPC Controller
0000:00:1F:2 Intel Corporation C610/X99 series chipset 6-Port SATA Controll
0000:00:1F:3 Intel Corporation C610/X99 series chipset SMBus Controller
0000:00:1F:6 Intel Corporation C610/X99 series chipset Thermal Subsystem
0000:01:00:0 Intel Corporation I350 Gigabit Network Connection (Interface e
0000:01:00:1 Intel Corporation I350 Gigabit Network Connection (Interface e
0000:03:00:0 Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (
0000:05:00:0 Intel Corporation Ethernet Controller XL710 for 40GbE QSFP+ (
0000:09:00:0 ASPEED Technology, Inc. AST1150 PCI-to-PCI Bridge
0000:0A:00:0 ASPEED Technology, Inc. ASPEED Graphics Family
0000:7F:08:0 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:08:2 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:08:3 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 QPI Link 0
0000:7F:0B:0 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 R3 QPI Link 0
0000:7F:0B:1 Intel Corporation Xeon E7 v3/Xeon E5 v3/Core i7 R3 QPI Link 0
```

6. Repeat steps **4** and **5** to add any additional Intel XL710 physical functions.

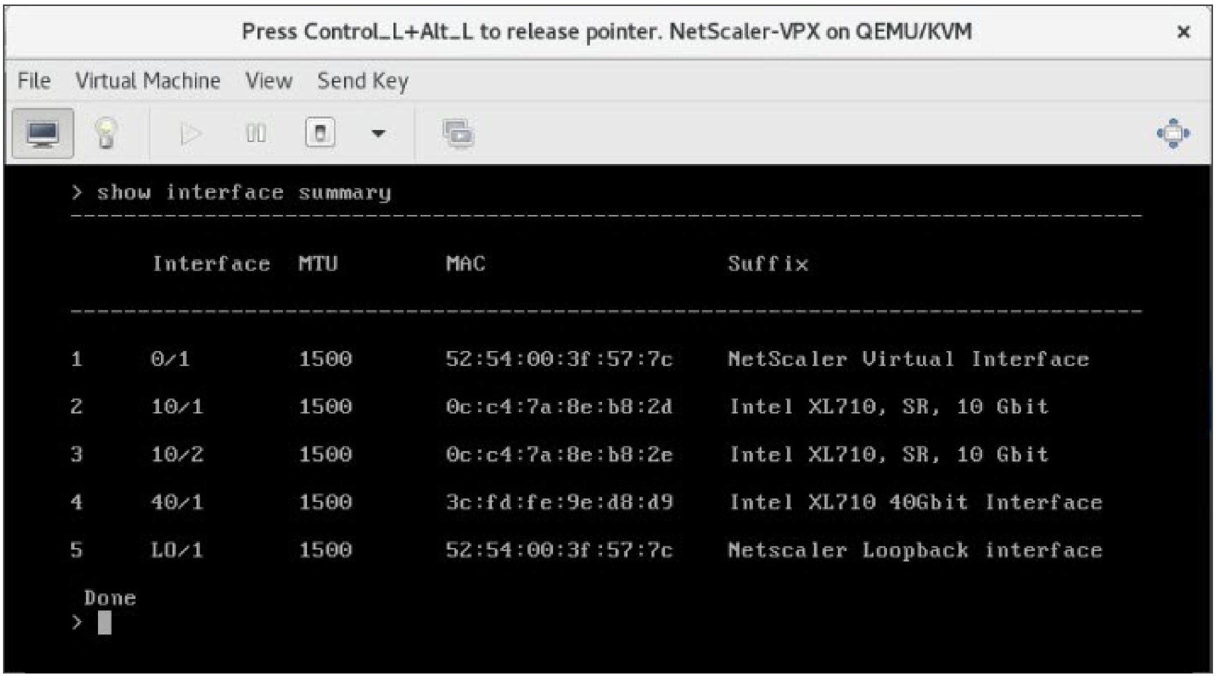
7. Power on the NetScaler VPX instance.

8. Once the NetScaler VPX instance powers on, you can use the following command to verify the configuration:

COMMAND

```
> show interface summary
```

The output should show all the interfaces that you configured:


Press Control_L+Alt_L to release pointer. NetScaler-VPX on QEMU/KVM
File
Virtual Machine
View
Send Key

```
> show interface summary
--------------------------------------------------------------------------------

      Interface  MTU        MAC                  Suffix

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

1     0/1        1500       52:54:00:3f:57:7c    NetScaler Virtual Interface

2     10/1       1500       0c:c4:7a:8e:b8:2d    Intel XL710, SR, 10 Gbit

3     10/2       1500       0c:c4:7a:8e:b8:2e    Intel XL710, SR, 10 Gbit

4     40/1       1500       3c:fd:fe:9e:d8:d9    Intel XL710 40Gbit Interface

5     LO/1       1500       52:54:00:3f:57:7c    Netscaler Loopback interface

 Done
>
```

Provision the NetScaler VPX instance by using the virsh program

September 21, 2020

The virsh program is a command line tool for managing VM Guests. Its functionality is similar to that of Virtual Machine Manager. It enables you to change a VM Guest's status (start, stop, pause, and so on), to set up new Guests and devices, and to edit existing configurations. The virsh program is also useful for scripting VM Guest management operations.

To provision NetScaler VPX by using the virsh program, follow these steps:

1. Use the tar command to untar the the NetScaler VPX package. The NSVPX-KVM-*_nc.tgz package contains following components:

   - The Domain XML file specifying VPX attributes [NSVPX-KVM-*_nc.xml]
   - Check sum of NS-VM Disk Image [Checksum.txt]
   - NS-VM Disk Image [NSVPX-KVM-*_nc.raw]

   Example:

   ```
   tar -xvzf NSVPX-KVM-10.1-117_nc.tgz
   NSVPX-KVM-10.1-117_nc.xml
   NSVPX-KVM-10.1-117_nc.raw
   checksum.txt
   <!--NeedCopy-->
   ```

2. Copy the NSVPX-KVM-*_nc.xml XML file to a file named <DomainName>-NSVPX-KVM-*_nc.xml. The <DomainName> is also the name of the virtual machine. Example:

```
cp NSVPX-KVM-10.1-117_nc.xml NetScaler-VPX-NSVPX-KVM-10.1-117_nc.
   xml
<!--NeedCopy-->
```

3. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to specify the following parameters:

   - name— Specify the name.
   - mac— Specify the MAC address.
     Note: The domain name and the MAC address have to be unique.
   - sourcefile— Specify the absolute disk-image source path. The file path has to be absolute. You can specify the path of the RAW image file or a QCOW2 image file.
     If you want to specify a RAW image file, specify the disk image source path as shown in the following example:

   Example:
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.raw'/>

   Specify the absolute QCOW2 disk-image source path and define the driver type as **qcow2**, as shown in the following example:

   Example:
<name>NetScaler-VPX</name>
<mac address='52:54:00:29:74:b3'/>
<driver name ='qemu' type='qcow2'/>
<source file='/root/NSVPX-KVM-10.1-117_nc.qcow'/>

4. Edit the <DomainName>-NSVPX-KVM-*_nc.xml file to configure the networking details:

   - source dev— specify the interface.
   - mode— specify the mode. The default interface is **Macvtap Bridge**.

   Example: Mode: MacVTap Bridge Set target interface as ethx and mode as bridge Model type as virtio

```
  <interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth0' mode='bridge'/>
      <target dev='macvtap0'/>
      <model type='virtio'/>
      <alias name='net0'/>
      <address type='pci' domain='0x0000' bus='0x00' slot='0x03'
         function='0x0'/>
```

```
    </interface>
<!--NeedCopy-->
```

Here, eth0 is the physical interface attached to the VM.

5. Define the VM attributes in the <DomainName>-NSVPX-KVM-*_nc.xml file by using the following command: virsh define <DomainName>-NSVPX-KVM-*_nc.xml Example:

```
virsh define NS-VPX-NSVPX-KVM-10.1-117_nc.xml
<!--NeedCopy-->
```

Start the VM by entering following command: virsh start [<DomainName>	<DomainUUID>] Example:

```
virsh start NetScaler-VPX
<!--NeedCopy-->
```

Connect the Guest VM through the console virsh console [<DomainName>	<DomainUUID>	<DomainID>] Example:

```
virsh console NetScaler-VPX
<!--NeedCopy-->
```

Add additional interfaces to NetScaler VPX instance using virsh program

After you have provisioned the NetScaler VPX on KVM, you can add additional interfaces.

To add additional interfaces, follow these steps:

1. Shut down the NetScaler VPX instance running on the KVM.

Edit the <DomainName>-NSVPX-KVM-*_nc.xml file using the command: virsh edit [<DomainName>	<DomainUUID>]

2.

3. In the <DomainName>-NSVPX-KVM-*_nc.xml file, append the following parameters:

a) **For MacVTap**

- Interface type— Specify the interface type as 'direct'.
- Mac address— Specify the Mac address and make sure the MAC address is unique across the interfaces.
- source dev— Specify the interface name.
- mode— Specify the mode; the modes supported are - Bridge, VEPA, Private, and Pass-through
- model type— Specify the model type as virtio

Example:

Mode: MacVTap Pass-through

Set target interface as
ethx, Mode as
bridge, and model type as
virtio

```
<interface type='direct'>
      <mac address='52:54:00:29:74:b3'/>
      <source dev='eth1' mode='passthrough'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

Here eth1 is the physical interface attached to the VM.

b) **For Bridge Mode**

Note: Make sure that you have configured a Linux bridge in the KVM host, bound the physical interface to the bridge, and put the bridge in the UP state.

- Interface type— Specify the interface type as 'bridge'.
- Mac address— Specify the Mac address and make sure the MAC address is unique across the interfaces.
- source bridge— Specify the bridge name.
- model type— Specify the model type as virtio

Example: Bridge Mode

```
<interface type='bridge'>
      <mac address='52:54:00:2d:43:a4'/>
      <source bridge='br0'/>
      <model type='virtio'/>
 </interface>
<!--NeedCopy-->
```

Manage the NetScaler VPX guest VMs

September 21, 2020

You can use the Virtual Machine Manager and the virsh program to perform management tasks such as starting or stopping a VM Guest, setting up new guests and devices, editing existing configurations, and connecting to the graphical console through Virtual Network Computing (VNC).

Manage the VPX guest VMs by using Virtual Machine Manager

- List the VM guests

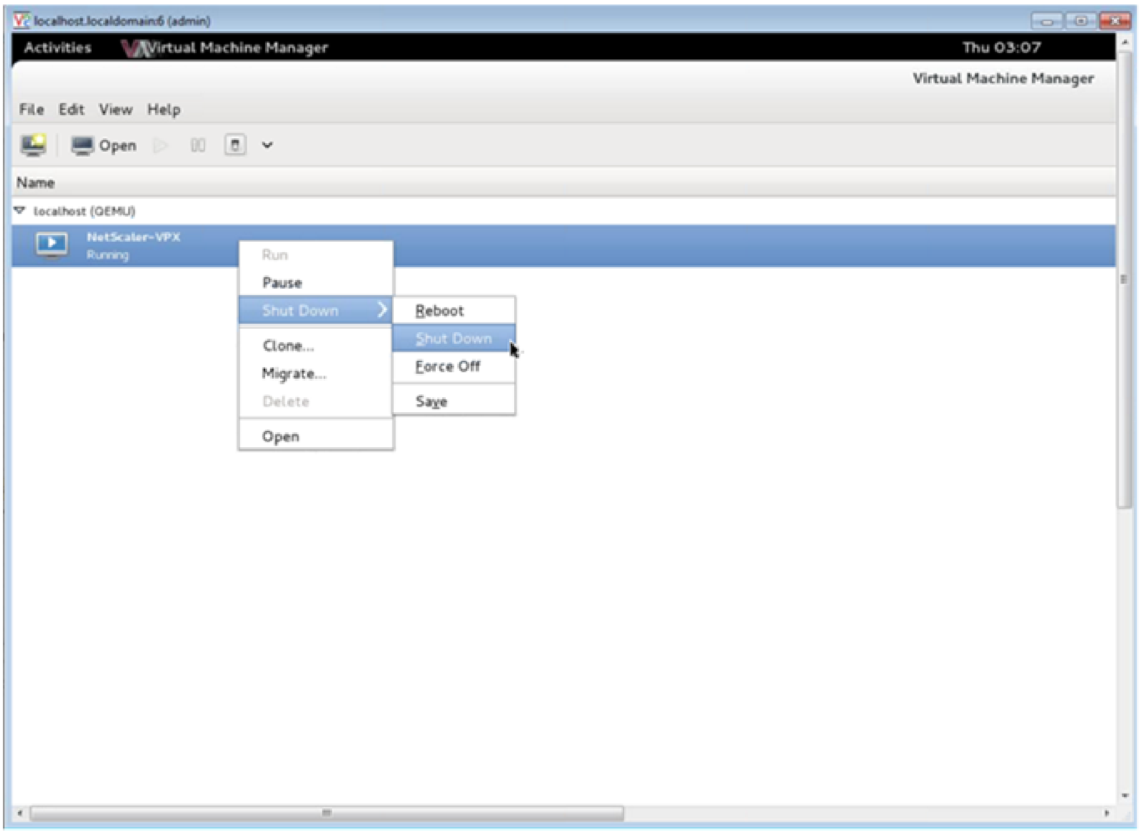
  The main Window of the Virtual Machine Manager displays a list of all the VM Guests for each VM host server it is connected to. Each VM Guest entry contains the virtual machine's name, along with its status (Running, Paused, or Shutoff) displayed as icon.

- Open a graphical console

  Opening a Graphical Console to a VM Guest enables you to interact with the machine like you would with a physical host through a VNC connection. To open the graphical console in the Virtual Machine Manager, right-click the VM Guest entry and select the Open option from the pop-up menu.

- Start and shut down a guest

  You can start or stop a VM Guest from the Virtual Machine Manager. To change the state of the VM, right-click the VM Guest entry and select Run or one of the Shut Down options from the pop-up menu.


localhost.localdomain6 (admin)
Activities
Virtual Machine Manager
Thu 03:07
Virtual Machine Manager
File Edit View Help
Open
Name
localhost (QEMU)
NetScaler-VPX
Running
Run
Pause
Shut Down
Clone...
Migrate...
Delete
Open
Reboot
Shut Down
Force Off
Save

- Reboot a guest

  You can reboot a VM Guest from the Virtual Machine Manager. To reboot the VM, right-click the VM Guest entry, and then select Shut Down > Reboot from the pop-up menu.

- Delete a guest

  Deleting a VM Guest removes its XML configuration by default. You can also delete a guest's storage files. Doing so completely erases the guest.

  1. In the Virtual Machine Manager, right-click the VM Guest entry.
  2. Select Delete from the pop-up menu. A confirmation window opens.
     Note: The Delete option is enabled only when the VM Guest is shut down.
  3. Click Delete.
  4. To completely erase the guest, delete the associated .raw file by selecting the Delete Associated Storage Files check box.

Manage the NetScaler VPX guest VMs using the virsh program

- List the VM Guests and their current states.

  To use virsh to display information about the Guests
  virsh list –all

The command output displays all domains with their states.
Example output:

```
Id Name                     State
----------------------------------
  0 Domain-0              running
  1 Domain-1              paused
  2 Domain-2              inactive
  3 Domain-3              crashed
<!--NeedCopy-->
```

- Open a virsh console.

  Connect the Guest VM through the console
  virsh console [<DomainID> | <DomainName> | <DomainUUID>]
  Example:
  virsh console NetScaler-VPX

- Start and shut down a guest.

  Guests can be started using the DomainName or Domain-UUID.
  virsh start [<DomainName> | <DomainUUID>]
  Example
  virsh start NetScaler-VPX

  To shut down a guest:
  virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
  Example:
  virsh shutdown NetScaler-VPX

- Reboot a guest

  virsh reboot [<DomainID> | <DomainName> | <DomainUUID>]
  Example:
  virsh reboot NetScaler-VPX

  Delete a guest

  To delete a Guest VM you need to shut-down the Guest and un-define the <DomainName>-NSVPX-KVM-*_nc.xml before you run the delete command.

```
virsh shutdown [<DomainID> | <DomainName> | <DomainUUID>]
virsh undefine [<DomainName> | <DomainUUID>]

<!--NeedCopy-->
```

  Example:

```
virsh shutdown NetScaler-VPX
virsh undefine NetScaler-VPX

<!--NeedCopy-->
```

Note: The delete command doesn't remove disk image file which needs to be removed manually.

Deploy a NetScaler VPX instance on AWS

September 21, 2020

You can launch a Citrix NetScaler VPX instance on Amazon Web Services (AWS). The NetScaler VPX appliance is available as an Amazon Machine Image (AMI) in AWS marketplace. A NetScaler VPX instance on AWS enables customer like you to leverage AWS Cloud computing capabilities and use NetScaler load balancing and traffic management features for their business needs. The VPX instance supports all the traffic management features of a physical NetScaler appliance, and they can be deployed as standalone instances or in HA pairs.

This section includes the following topics:

- AWS terminology
- How a NetScaler VPX instance on AWS works
- Supported instance type, ENI, and IP addresses

AWS terminology

Here is a brief description of the terms used in this document. For more information, see AWS Glossary.

Term	Defintion
Amazon Machine Image (AMI)	A machine image, which provides the information required to launch an instance, which is a virtual server in the cloud.
Elastic Block Store	Provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.
Simple Storage Service (S3)	Storage for the Internet. It is designed to make web-scale computing easier for developers.

Term	Defintion
Elastic Compute Cloud (EC2)	A web service that provides secure, resizable compute capacity in the cloud. It is designed to make web-scale cloud computing easier for developers.
Elastic Load Balancing (ELB)	Distributes incoming application traffic across multiple EC2 instances, in multiple Availability Zones. This increases the fault tolerance of your applications.
Elastic network interface (ENI)	A virtual network interface that you can attach to an instance in a VPC.
Elastic IP (EIP) address	A static, public IPv4 address that you have allocated in Amazon EC2 or Amazon VPC and then attached to an instance. Elastic IP addresses are associated with your account, not a specific instance. They are elastic because you can easily allocate, attach, detach, and free them as your needs change.
Instance type	Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.
Identity and Access Management (IAM)	An AWS identity with permission policies that determine what the identity can and cannot do in AWS. You can use an IAM role to enable applications running on an EC2 instance to securely access your AWS resources.IAM role is required for deploying VPX instances in a high-availability setup.
Internet Gateway	Connects a network to the Internet. You can route traffic for IP addresses outside your VPC to the Internet gateway.

Term	Defintion
Key pair	A set of security credentials that you use to prove your identity electronically. A key pair consists of a private key and a public key.
Route tables	A set of routing rules that controls the traffic leaving any subnet that is associated with the route table. You can associate multiple subnets with a single route table, but a subnet can be associated with only one route table at a time.
Security groups	A named set of allowed inbound network connections for an instance.
Subnets	A segment of the IP address range of a VPC that EC2 instances can be attached to. You can create subnets to group instances according to security and operational needs.
Virtual Private Cloud (VPC)	A web service for provisioning a logically isolated section of the AWS cloud where you can launch AWS resources in a virtual network that you define.
Auto Scaling	A web service to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks.
CloudFormation	A service for writing or changing templates that create and delete related AWS resources together as a unit.

How a NetScaler VPX instance on AWS works

The NetScaler VPX instance is available as an AMI in AWS marketplace, and it can be launched as an EC2 instance within an AWS VPC. The NetScaler VPX AMI instance requires a minimum of 2 virtual CPUs and 2 GB of memory. An EC2 instance launched within an AWS VPC can also provide the multiple interfaces, multiple IP addresses per interface, and public and private IP addresses needed for VPX configuration. Each VPX instance requires at least three IP subnets:

- A management subnet
- A client-facing subnet (VIP)

- A back-end facing subnet (SNIP,MIP, etc.)

Citrix recommends three network interfaces for a standard VPX instance on AWS installation.

AWS currently makes multi-IP functionality available only to instances running within an AWS VPC. A VPX instance in a VPC can be used to load balance servers running in EC2 instances. An Amazon VPC allows you to create and control a virtual networking environment, including your own IP address range, subnets, route tables, and network gateways.

Note: By default, you can create up to 5 VPC instances per AWS region for each AWS account. You can request higher VPC limits by submitting Amazon's request form http://aws.amazon.com/contact-us/vpc-request.

Figure 1. A Sample NetScaler VPX Instance Deployment on AWS Architecture

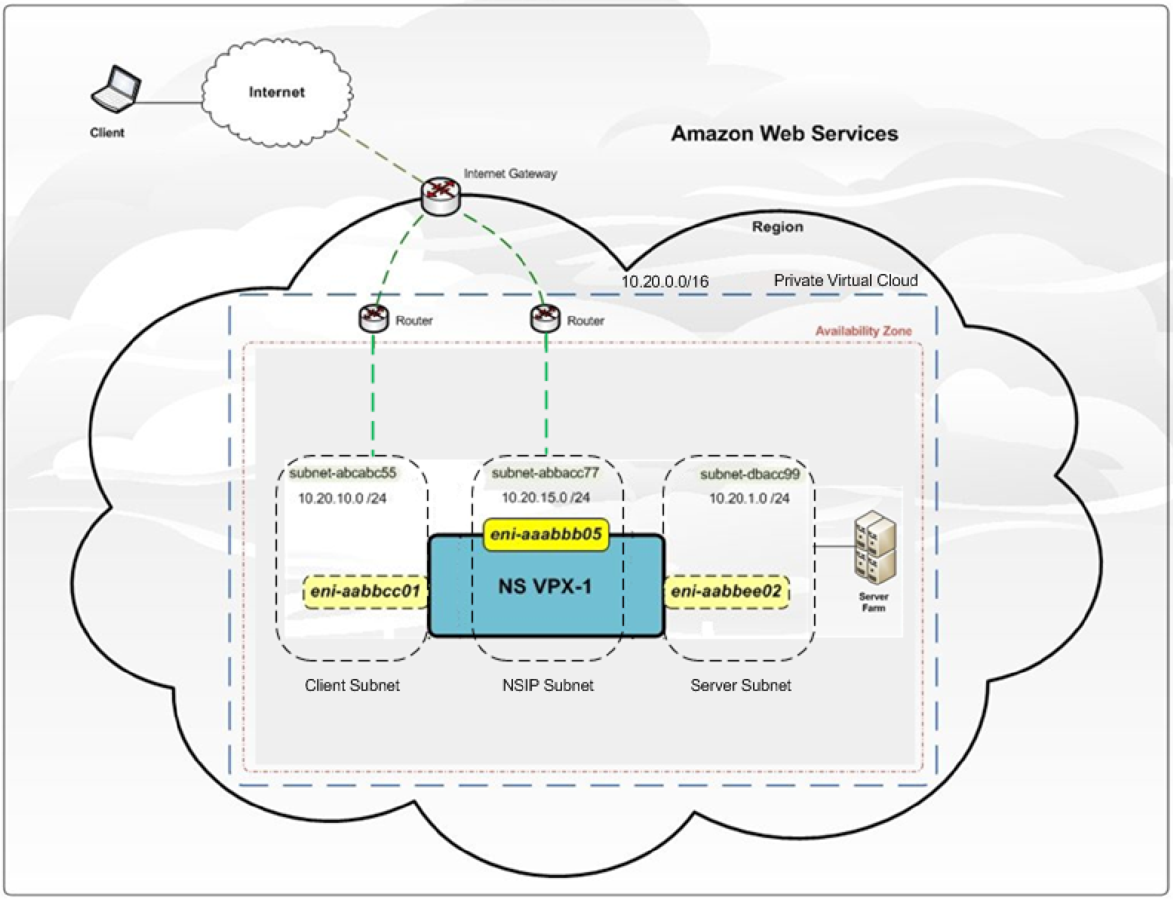

Client
Internet
Amazon Web Services
Internet Gateway
Region
10.20.0.0/16
Private Virtual Cloud
Router
Router
Availability Zone
subnet-abcabc55
10.20.10.0 /24
subnet-abbacc77
10.20.15.0 /24
subnet-dbacc99
10.20.1.0 /24
eni-aaabbb05
eni-aabbcc01
NS VPX-1
eni-aabbee02
Server Farm
Client Subnet
NSIP Subnet
Server Subnet

Figure 1 shows a simple topology of an AWS VPC with a NetScaler VPX deployment. The AWS VPC has:

1. A single Internet gateway to route traffic in and out of the VPC.
2. Network connectivity between the Internet gateway and the Internet.
3. Three subnets, one each for management, client, and server.

4. Network connectivity between the Internet gateway and the two subnets (management and client).
5. A standalone NetScaler VPX instance deployed within the VPC. The VPX instance has three ENIs, one attached to each subnet.

Supported instance type, ENI, and IP addresses

For more information about Amazon EC2 instances and IP addresses supported per NIC per instance type:

- Instance types
- IP addresses per network interface per instance type

For higher bandwidth, Citrix recommends the following instance types:

Instance type	Bandwidth	Enhanced networking (SR-IOV)
M4.10x large	3 Gbps and 5 Gbps	Yes
C4.8x large	3 Gbps and 5 Gbps	Yes

Limitations and usage guidelines

September 21, 2020

The following limitations and usage guidelines apply when deploying a NetScaler VPX instance on AWS:

- Before you start, read the AWS terminology section in Deploy a NetScaler VPX instance on AWS.
- The clustering feature is not supported for VPX.
- For the high availability setup to work effectively, associate a dedicated NAT device to management Interface or associate EIP to NSIP. For more information on NAT, in the AWS documentation, see NAT Instances.
- Data traffic and management traffic must be segregated with ENIs belonging to different subnets.
- Only the NSIP address must be present on the management ENI.
- If a NAT instance is used for security instead of assigning an EIP to the NSIP, appropriate VPC level routing changes are required. For instructions on making VPC level routing changes, in the AWS documentation, see Scenario 2: VPC with Public and Private Subnets.

- A VPX instance can be moved from one EC2 instance type to another (for example, from m3.large to an m3.xlarge).
- For storage options for VPX on AWS, Citrix recommends EBS, because it is durable and the data is available even after it is detached from instance.
- Dynamic addition of ENIs to VPX is not supported. Restart the VPX instance to apply the update. Citrix recommends you to stop the standalone or HA instance, attach the new ENI, and then restart the instance.
- You can assign multiple IP addresses to an ENI. The maximum number of IP addresses per ENI is determined by the EC2 instance type, see the section "IP Addresses Per Network Interface Per Instance Type" in Elastic Network Interfaces. You must allocate the IP addresses in AWS before you assign them to ENIs. For more information, see Elastic Network Interfaces.
- Citrix recommends that you avoid using the enable and disable interface commands on NetScaler VPX interfaces.
- The NetScaler `set ha node \<NODE\\_ID\> -haStatus STAYPRIMARY` and `set ha node \<NODE\\_ID\> -haStatus STAYSECONDARY` commands are disabled by default.
- IPv6 is not supported for VPX.
- Due to AWS limitations, these features are not supported:
  - Gratuitous ARP(GARP)
  - L2 mode
  - Tagged VLAN
  - Dynamic Routing
  - Virtual MAC
- For RNAT to work, ensure **Source/Destination Check is** disabled. For more information, see "Changing the Source/Destination Checking" in Elastic Network Interfaces.
- In a NetScaler VPX deployment on AWS, in some AWS regions, the AWS infrastructure might not be able to resolve AWS API calls. This happens if the API calls are issued through a nonmanagement interface on the NetScaler VPX instance.
  As a workaround, restrict the API calls to the management interface only. To do that, create an NSVLAN on the VPX instance and bind the management interface to the NSVLAN by using the appropriate command.
  For example:
  set ns config -nsvlan <vlan id> -ifnum 1/1 -tagged NO
  save config
  Restart the VPX instance at the prompt. For more information about configuring NSVLAN, see Configuring NSVLAN.

- In the AWS console, the vCPU usage shown for a VPX instance under the **Monitoring** tab might be high (up to 100 percent), even when the actual usage is much lower. To see the actual vCPU usage, use the VPX GUI or CLI.

Deploy a NetScaler VPX standalone instance on AWS

September 21, 2020

You can deploy a Citrix
NetScaler VPX standalone instance on AWS by using the following options:

1. AWS web console
2. Citrix-authored CloudFormation template
3. AWS CLI

This topic describes the procedure for deploying a NetScaler VPX instance on AWS.

Before you start your deployment, read the following topics:

- Prerequisites
- Limitation and usage guidelines

Deploy a NetScaler VPX instance on AWS by using the AWS web console

You can deploy a NetScaler VPX instance on AWS through the AWS web console. The deployment process includes the following steps:

1. Create a Key Pair
2. Create a Virtual Private Cloud (VPC)
3. Add additional subnets
4. Create security groups and security rules
5. Add route tables
6. Create an internet gateway
7. Create a NetScaler VPX instance
8. Create and attach additional network interfaces
9. Attach elastic IPs to the management NIC
10. Connect to the VPX instance

Step 1: Create a key pair.

Amazon EC2 uses a key pair to encrypt and decrypt logon information. To log on to your instance, you must create a key pair, specify the name of the key pair when you launch the instance, and provide the private key when you connect to the instance.

When you review and launch an instance by using the AWS Launch Instance wizard , you are prompted to use an existing key pair or create a new key pair. More more information about how to create a key pair, see Amazon EC2 Key Pairs.

Step 2: Create a VPC.

A NetScaler VPC instance is deployed inside an AWS VPC. A VPC allows you to define virtual network dedicated to your AWS account. For more information about AWS VPC, see Getting Started With Amazon VPC.

While creating a VPC for your NetScaler VPX instance, keep the following points in mind.

- Use the VPC with a Single Public Subnet Only option to create a new AWS VPC in an AWS availability zone.
- Citrix recommends that you create at least **three subnets**, of the following types:
  - One subnet for management traffic. You place the management IP(NSIP) on this subnet. By default elastic network interface (ENI) eth0 is used for management IP.
  - One or more subnets for client-access (user-to-NetScaler VPX) traffic, through which clients connect to one or more virtual IP (VIP) addresses assigned to NetScaler load balancing virtual servers.
  - One or more subnets for the server-access (VPX-to-server) traffic, through which your servers connect to VPX-owned subnet IP (SNIP) addresses. For more information about NetScaler load balancing and virtual servers, virtual IP addresses (VIPs), and subnet IP addresses (SNIPs), see:
  - All subnets must be in the same availability zone.

Step 3: Add subnets.

When you used the VPC wizard, only one subnet was created. Depending on your requirement, you might want to create additional subnets. For more information about how to create additional subnets, see Adding a Subnet to Your VPC.

Step 4: Create security groups and security rules.

To control inbound and outbound traffic, create security groups and add rules to the groups. For more information how to create groups and add rules, see Security Groups for Your VPC.

For NetScaler VPX instances, the EC2 wizard gives default security groups, which are generated by AWS Marketplace and is based on recommended settings by Citrix. However, you can create additional security groups based on your requirements.

> **Note**
>
> Port 22, 80, 443 to be opened on the Security group for SSH, HTTP, and HTTPS access respectively.

Step 5: Add route tables.

Route table contains a set of rules, called routes, that are used to determine where network traffic is directed. Each subnet in your VPC must be associated with a route table. For more information about how to create a route table, see Route Tables.

Step 6: Create an internet gateway.

An internet gateway serves two purposes: to provide a target in your VPC route tables for internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

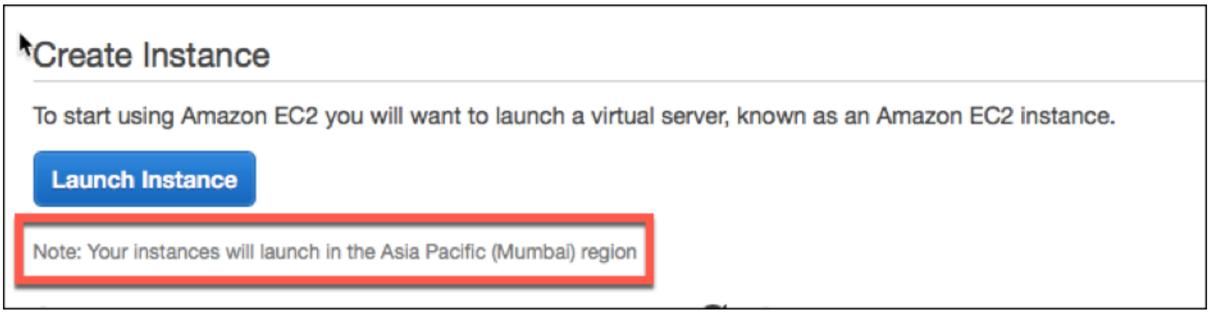
Create an internet gateway for internet traffic. For more information about how to create an Internet Gateway, see the section Attaching an Internet Gateway.

Step 7: Create a NetScaler VPX instance by using the AWS EC2 service.

To create a NetScaler VPX instance by using the AWS EC2 service, complete the following steps.

1. From the AWS dashboard, go to Compute > EC2 > Launch Instance > AWS Marketplace.

Before you click **Launch Instance**, ensure your region is correct by checking the note that appears under Launch Instance.



Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Note: Your instances will launch in the Asia Pacific (Mumbai) region

2. In the Search AWS Marketplace bar, search with the keyword NetScaler VPX.

3. Select the version you want to deploy and then click **Select**. For the NetScaler VPX version, you the following options:

- A licensed version
- NetScaler VPX Express appliance (This is a free virtual appliance, which is available from NetScaler 12.0 56.20.)
- Bring your own device

The Launch Instance wizard starts. Follow the wizard to create an instance. The wizard prompts you to:

- Choose Instance Type
- Configure Instance
- Add Storage
- Add Tags
- Configure Security Group

- Review

Services
Resource Groups
1. Choose AMI
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review

Step 8: Create and attach additional network interfaces.

Create two additional network interfaces for VIP and SNIP. For more information about how to create additional network interfaces, see the Creating a Network Interface section.

After you've created the network interfaces, you must attach them to the VPX instance. Before attaching the interface, shut down the VPX instance, attach the interface, and power on the instance. For more information about how to attach network interfaces, see the Attaching a Network Interface When Launching an Instance section.

Step 9: Allocate and associate elastic IPs.

If you assign a public IP address to an EC2 instance, it remains assigned only until the instance is stopped. After that, the address is released back to the pool. When you restart the instance, a new public IP address is assigned.

In contrast, an elastic IP (EIP) address remains assigned until the address is disassociated from an instance.

Allocate and associate an elastic IP for the management NIC. For more information about how to allocate and associate elastic IP addresses, see these topics:

- Allocating an Elastic IP Address
- Associating an Elastic IP Address with a Running Instance

These steps complete the procedure to create a NetScaler VPX instance on AWS. It can take a few minutes for the instance to be ready. Check that your instance has passed its status checks. You can view this information in the Status Checks column on the Instances page.

Step 10: Connect to the VPX instance.

After you've created the VPX instance, you connect the instance by using the GUI and an SSH client.

- GUI

The following are the default administrator credentials to access a NetScaler VPX instance

User name: nsroot

Password: The default password for the nsroot account is set to the AWS instance-ID of the NetScaler VPX instance.

- SSH client

From the AWS management console, select the NetScaler VPX instance and click **Connect**. Follow the instructions given on the **Connect to Your Instance** page.

For more information about how to deploy a NetScaler VPX standalone instance on AWS by using the AWS web console, see:

- Scenario: standalone instance
- How to configure a Citrix NetScaler VPX instance on AWS by using Citrix CloudFormation template

Configure a NetScaler VPX instance by using the Citrix CloudFormation template

You can use the Citrix-provided CloudFormation template to automate VPX instance launch. The template provides functionality to launch a single NetScaler VPX instance, or to create a high availability environment with a pair of NetScaler VPX instances.

You can launch the template from AWS Marketplace or GitHub.

The CloudFormation template requires an existing VPC environment, and it launches a VPX instance with three elastic network interfaces (ENIs). Before you start the CloudFormation template, ensure that you complete the following requirements:

- An AWS virtual private cloud (VPC)
- Three subnets within the VPC: one for management, one for client traffic, and one for back-end servers
- An EC2 key pair to enable SSH access to the instance
- A security group with UDP 3003, TCP 3009-3010, HTTP, SSH ports open

See the "Deploy a NetScaler VPX Instance on AWS by Using the AWS Web Console" section or AWS documentation for more information about how to complete the prerequisites.

Watch this video to learn about how to configure and launch a NetScaler VPX standalone instance by using the Citrix CloudFormation template available in the AWS Marketplace.

Further, you configure and launch a NetScaler VPX Express standalone instance by using the Citrix CloudFormation template available in GitHub:

https://github.com/citrix/netscaler-aws-cloudformation

An IAM role is not mandatory for a standalone deployment. However, Citrix recommends that you create and attach an IAM role with the required privileges to the instance, for future need. The IAM role ensures that the standalone instance is easily converted to a high availability node with SR-IOV, when required.

For more information about the required privileges, see Configuring NetScaler VPX instances to Use SR-IOV Network Interface.

Configure a NetScaler VPX instance by using the AWS CLI

You can use the AWS CLI to launch instances. For more information, see the AWS Command Line Interface Documentation.

Download a NetScaler VPX license

September 21, 2020

After the launch of Citrix ADC VPX-customer licensed instance from the AWS marketplace, a license is required. For more information on VPX licensing, see Licensing overview.

You have to:

1. Use the licensing portal within Citrix website to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance will activate automatically.

If you use a NetScaler VPX instance with a model number higher than VPX 5000, the network throughput might not be the same as specified by the instance's license. However, other features, such as SSL throughput and SSL transactions per second, might improve.

5 Gbps network bandwidth is observed in c4.8xlarge instance type.

Load balancing servers in different availability zones

September 21, 2020

A VPX instance can be used to load balance servers running in the same availability zone, or in:

- A different availability zone (AZ) in the same AWS VPC
- A different AWS region
- AWS EC2 in a VPC

To enable a VPX instance to load balance servers running outside the AWS VPC that the VPX instance is in, configure the instance to use EIPs to route traffic through the Internet gateway, as follows:

1. Configure a SNIP on the NetScaler VPX instance by using the NetScaler CLI or the GUI.
2. Enable traffic to be routed out of the AZ, by creating a public facing subnet for the server-side traffic.

3. Add an Internet gateway route to the routing table, using the AWS GUI console.
4. Associate the routing table you just updated with the server-side subnet.
5. Associate an EIP with the server-side private IP address that is mapped to a NetScaler SNIP address.

Deploy a high availability pair on AWS

September 21, 2020

You can configure two NetScaler VPX instances on AWS as a high availability active-passive pair. With one instance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

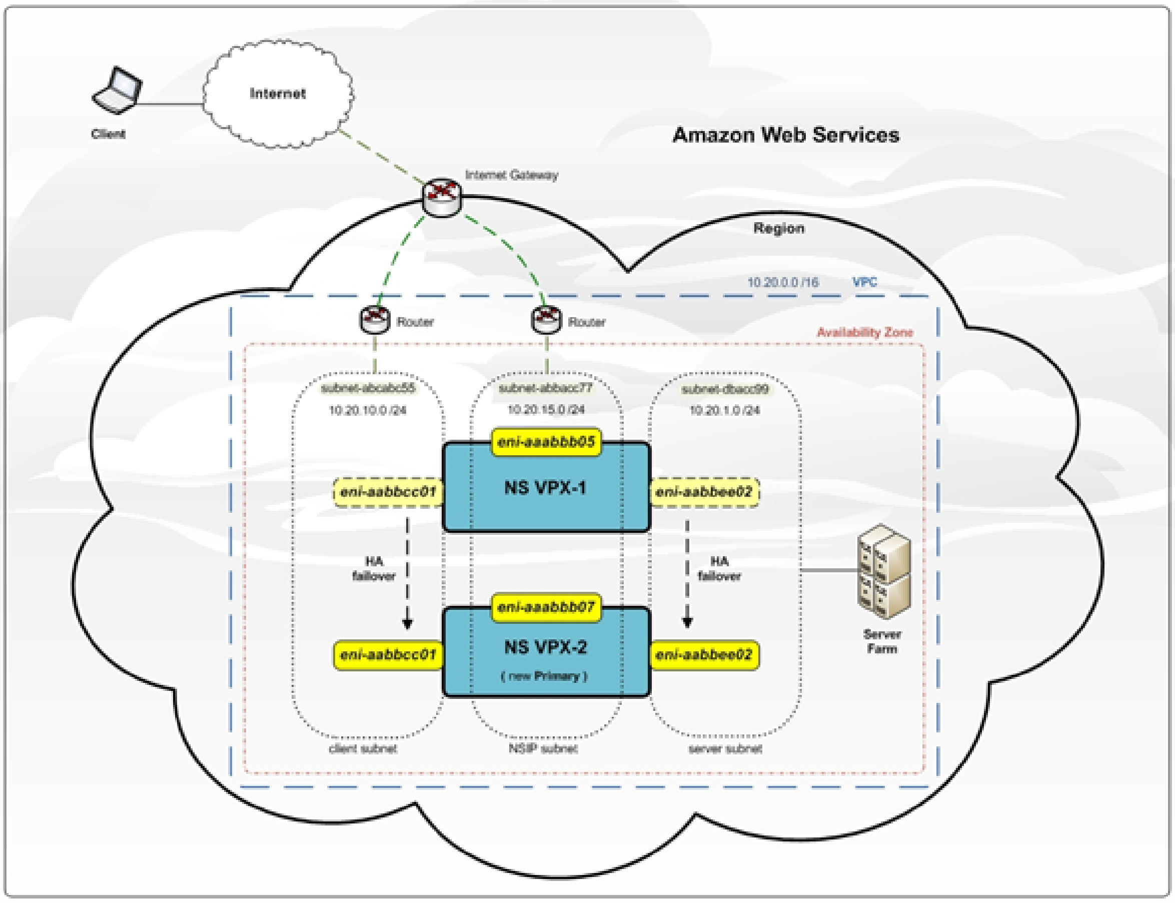
For more information on HA, see High availability.

Before you start your deployment, read the following topics:

- Prerequisites for the required IAM role privileges.
- Limitations and usage guidelines

The following figure shows an example of the HA deployment architecture for
NetScaler VPX instances on AWS.

Figure 1. A NetScaler VPX HA Pair on AWS


Internet
Client
Amazon Web Services
Internet Gateway
Region
10.20.0.0 /16
VPC
Router
Router
Availability Zone
subnet-abcabc55
10.20.10.0 /24
subnet-abbacc77
10.20.15.0 /24
subnet-dbacc99
10.20.1.0 /24
eni-aaabbb05
eni-aabbcc01
NS VPX-1
eni-aabbee02
HA failover
HA failover
Server Farm
eni-aaabbb07
eni-aabbcc01
NS VPX-2
(new Primary)
eni-aabbee02
client subnet
NSIP subnet
server subnet

To deploy HA for two VPX instances on AWS, you either create the instances with IAM Role manually by using the AWS Management Console and then configure HA on them, or you can automate the HA deployment by using the Citrix CloudFormation template.

The CloudFormation template significantly decreases the number of steps involved for creating an HA pair, and it automatically creates an IAM Role. This section shows how to deploy a NetScaler VPX HA (active-passive) pair by using the Citrix CloudFormation template.

Keep the following points in mind while deploying two NetScaler VPX instances as an HA pair.

Points to note

- HA on AWS requires the primary node to have at least two ENIs (one for management and the other for data traffic), and the secondary node to have one management ENI. However, for security purposes, create three ENIs on the primary node, because this setup allows you to segregate private and public network (recommended).
- The secondary node always has one ENI interface (for management) and the primary node can have up to four ENIs.

- The NSIP addresses for each VPX instance in a high availability pair must be configured on the default ENI of the instance.
- Because Amazon does not allow any broadcast/multicast packets in AWS, HA is implemented by migrating data-plane ENIs from the primary to the secondary (new primary) VPX instance when the primary VPX instance fails.
- Because the default (management) ENI cannot be moved to another VPX instance, do not use the default ENI for client and server traffic (data-plane traffic).
- The message AWSCONFIG IOCTL NSAPI_HOTPLUG_INTF success output 0 in the /var/log/ns.log indicates that the two data ENIs have successfully attached to the secondary instance (the new primary).
- Failover might take up to 20 seconds due to the AWS detach/attach ENI mechanism.
- Upon failover, the failed instance always restarts.
- The heartbeat packets are received only on the management interface.
- The configuration file of the primary and secondary VPX instances is synchronized, including the `nsroot` password. The `nsroot` password of the secondary node is set to that of the primary node after the HA configuration synchronization.
- To have access to the AWS API servers, either the VPX instance must have a public IP address assigned or routing must be set up correctly at VPC subnet level pointing to internet gateway of the VPC
- Nameservers/DNS servers are configured at VPC level using DHCP options.
- The Citrix CloudFormation template does not create an HA setup between different availability zones.
- The Citrix CloudFormation template does not create an INC mode.
- The AWS debug messages are available in the log file, /var/log/ns.log, on the VPX instance.

Deploy a high availability pair by using the Citrix CloudFormation template

Before starting the CloudFormation template, ensure that you complete the following requirements:

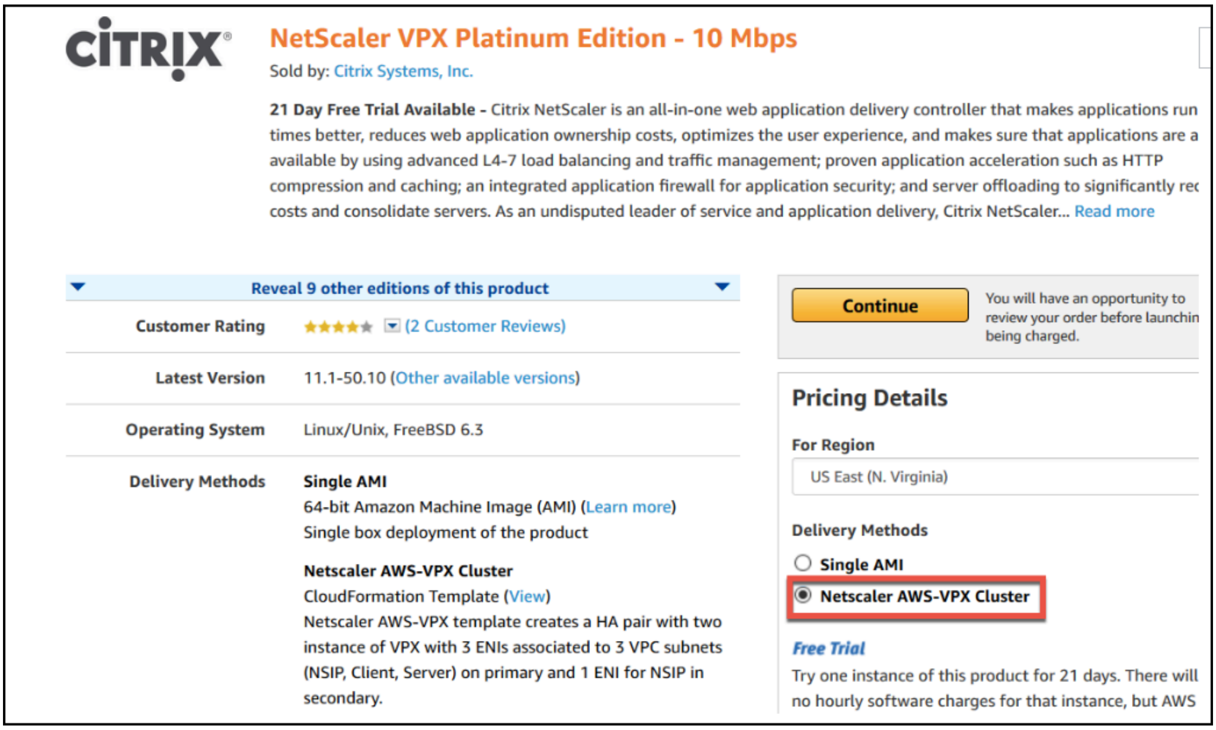
- A VPC
- Three subnets within the VPC
- A security group with UDP 3003, TCP 3009–3010, HTTP, SSH ports open
- A key pair

> **Note**
>
> The Citrix CloudFormation template automatically creates an IAM Role. The template has no option to select an existing IAM Role.

To launch the Citrix CloudFormation template:

1. Log on to the AWS marketplace by using your AWS credentials.
2. In the search field, type **NetScaler VPX** to search for the NetScaler AMI, and click **Go**.

3. On the search result page, click the desired NetScaler VPX offering.
4. Under **For Region**, select your region.
5. Select the **Delivery Methods** as **Netscaler AWS-VPX Cluster** and click **Continue**.



NetScaler VPX Platinum Edition - 10 Mbps

Sold by: Citrix Systems, Inc.

21 Day Free Trial Available - Citrix NetScaler is an all-in-one web application delivery controller that makes applications run times better, reduces web application ownership costs, optimizes the user experience, and makes sure that applications are a available by using advanced L4-7 load balancing and traffic management; proven application acceleration such as HTTP compression and caching; an integrated application firewall for application security; and server offloading to significantly rec costs and consolidate servers. As an undisputed leader of service and application delivery, Citrix NetScaler... Read more

Customer Rating	★★★★★ (2 Customer Reviews)
Latest Version	11.1-50.10 (Other available versions)
Operating System	Linux/Unix, FreeBSD 6.3
Delivery Methods	**Single AMI** 64-bit Amazon Machine Image (AMI) (Learn more) Single box deployment of the product **Netscaler AWS-VPX Cluster** CloudFormation Template (View) Netscaler AWS-VPX template creates a HA pair with two instance of VPX with 3 ENIs associated to 3 VPC subnets (NSIP, Client, Server) on primary and 1 ENI for NSIP in secondary.

You will have an opportunity to review your order before launchin being charged.

Pricing Details

For Region

US East (N. Virginia)

Delivery Methods

○ **Single AMI**

● **Netscaler AWS-VPX Cluster**

Free Trial

Try one instance of this product for 21 days. There will no hourly software charges for that instance, but AWS

6. On the **Launch on EC2** page, under **Version**, select the correct NetScaler. Ensure that the **Region** and **Deployment Options** are correct. Check pricing details.

7. Click **Launch with CloudFormation Console** to launch the Citrix CloudFormation template.

8. The **Select Template** page appears. Click **Next**.

9. The **Specify Details** appears. Enter the following details.

a. Type a **Stack name.** The name must be within 25 characters.

b. Under **High Availability Configuration**

Select **Yes** from the menu for **Create HA pair?**.

c. Under **Virtual Private Network Configuration**

Select the VPC that you've already created for **VPC ID.**

Type **Remote SSH CIDR IP**.

Type **Remote HTTP CIDR IP**.

Type **Remote HTTPS CIDR IP.**

Select the key pair that you've already created from the drop-down menu for **Key Pair**.

d. Under **Network Interface Configuration**

Select **Management Subnetwork**, **Client Subnetwork**, and **Server Subnetwork**. Ensure that you select the correct subnetworks you created within the VPC that you selected under VPC ID in step c.

Add **Primary Management IP**, **Secondary Management IP**, **Client IP**, and **Server IP.** The IP addresses must belong to the same subnets of the respective subnetworks. Alternatively, you can let the template assign the IP addresses automatically.

e. Under **Other Parameters**

Select **m4.large** for **Instant Type**.

Select **default** for **Tenancy Type**.

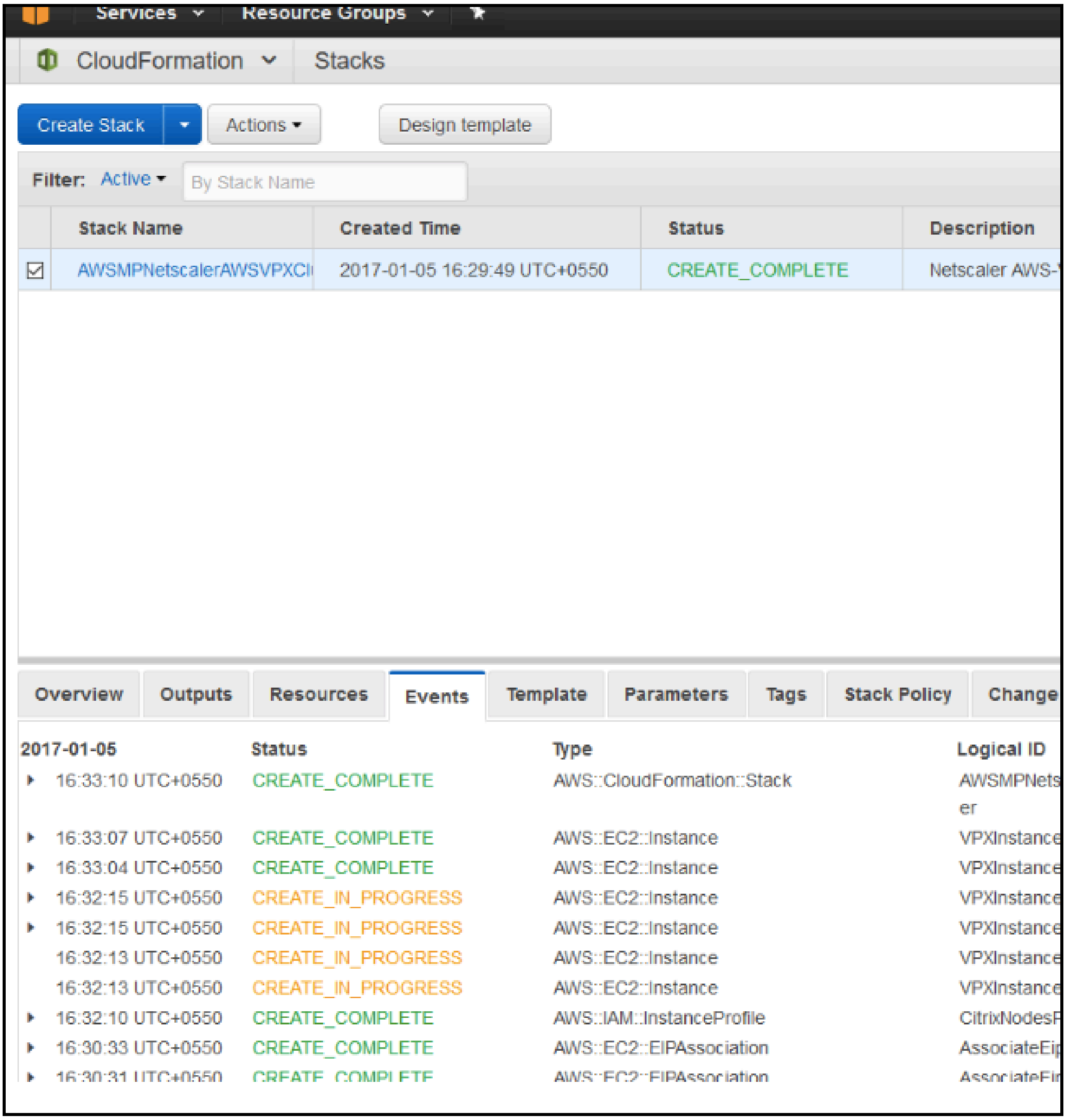
f. Click **Next**.

8. The **Options** page appears. (This is an optional page.). Click **Next**.

9. The **Review** page appears. Take a moment to review the settings and make necessary changes if necessary.

10. Select the **I acknowledge that AWS CloudFormation might create IAM resources.** check box, and then click **Create**.

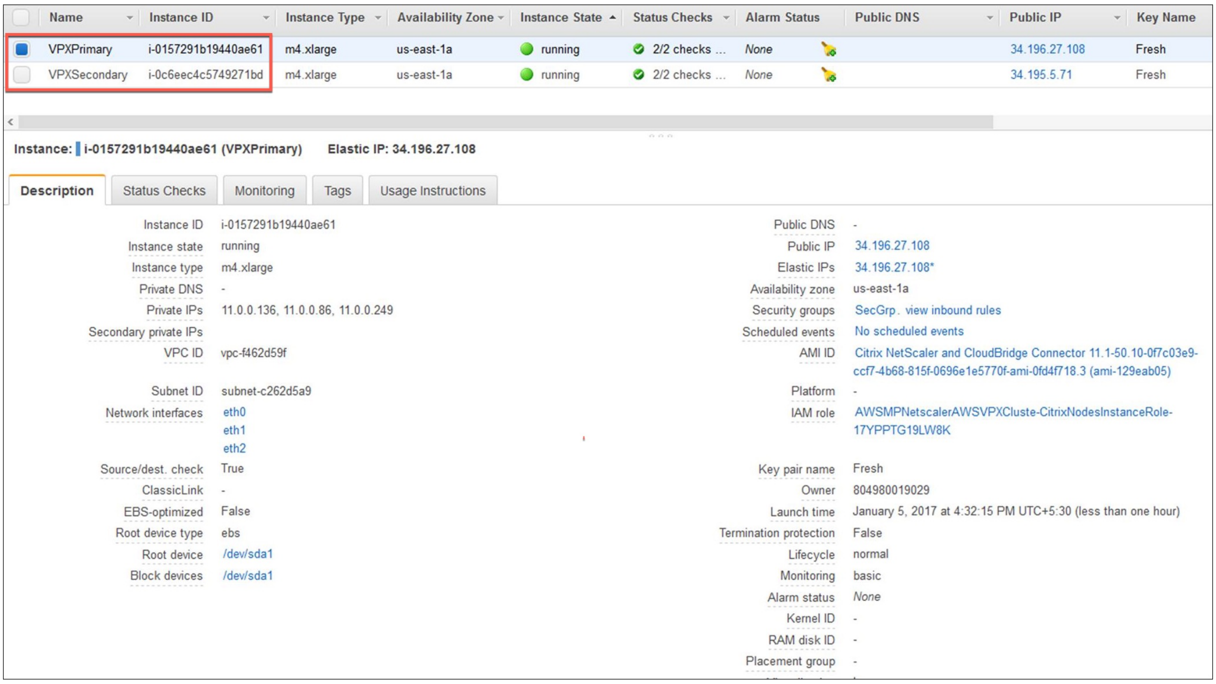
11. The **CREATE-IN-PROGRESS** status appears. Wait until the status is **CREATE-COMPLETE**. If the status does not change to "COMPLETE," check the **Events** tab for the reason of failure and recreate the instance with proper configurations.


Services
Resource Groups
CloudFormation
Stacks
Create Stack
Actions
Design template
Filter: Active
By Stack Name
Overview
Outputs
Resources
Events
Template
Parameters
Tags
Stack Policy
Change

	Stack Name	Created Time	Status	Description
☑	AWSMPNetscalerAWSVPXCl	2017-01-05 16:29:49 UTC+0550	CREATE_COMPLETE	Netscaler AWS-

2017-01-05	Status	Type	Logical ID
16:33:10 UTC+0550	CREATE_COMPLETE	AWS::CloudFormation::Stack	AWSMPNets er
16:33:07 UTC+0550	CREATE_COMPLETE	AWS::EC2::Instance	VPXInstance
16:33:04 UTC+0550	CREATE_COMPLETE	AWS::EC2::Instance	VPXInstance
16:32:15 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	VPXInstance
16:32:15 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	VPXInstance
16:32:13 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	VPXInstance
16:32:13 UTC+0550	CREATE_IN_PROGRESS	AWS::EC2::Instance	VPXInstance
16:32:10 UTC+0550	CREATE_COMPLETE	AWS::IAM::InstanceProfile	CitrixNodesF
16:30:33 UTC+0550	CREATE_COMPLETE	AWS::EC2::EIPAssociation	AssociateEip
16:30:31 UTC+0550	CREATE_COMPLETE	AWS::EC2::EIPAssociation	AssociateEi

11. After an IAM resource is created, go to EC2 **Management Console > Instances**. You notice two VPX instances created with IAM role. The primary node is created with three private IP addresses and three network interfaces.



☐	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name
☑	VPXPrimary	i-0157291b19440ae61	m4.xlarge	us-east-1a	● running	✓ 2/2 checks ...	None		34.196.27.108	Fresh
☐	VPXSecondary	i-0c6eec4c5749271bd	m4.xlarge	us-east-1a	● running	✓ 2/2 checks ...	None		34.195.5.71	Fresh

Instance: i-0157291b19440ae61 (VPXPrimary) Elastic IP: 34.196.27.108

Instance ID	i-0157291b19440ae61	Public DNS	-
Instance state	running	Public IP	34.196.27.108
Instance type	m4.xlarge	Elastic IPs	34.196.27.108*
Private DNS	-	Availability zone	us-east-1a
Private IPs	11.0.0.136, 11.0.0.86, 11.0.0.249	Security groups	SecGrp. view inbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-f462d59f	AMI ID	Citrix NetScaler and CloudBridge Connector 11.1-50.10-0f7c03e9-ccf7-4b68-815f-0696e1e5770f-ami-0fd4f718.3 (ami-129eab05)
Subnet ID	subnet-c262d5a9	Platform	-
Network interfaces	eth0 eth1 eth2	IAM role	AWSMPNetscalerAWSVPXCluste-CitrixNodesInstanceRole-17YPPTG19LW8K
Source/dest. check	True	Key pair name	Fresh
ClassicLink	-	Owner	804980019029
EBS-optimized	False	Launch time	January 5, 2017 at 4:32:15 PM UTC+5:30 (less than one hour)
Root device type	ebs	Termination protection	False
Root device	/dev/sda1	Lifecycle	normal
Block devices	/dev/sda1	Monitoring	basic
		Alarm status	None
		Kernel ID	-
		RAM disk ID	-
		Placement group	-

The secondary node is created with one private IP address and one network interface.

> **Note**
>
> The secondary node is created with one interface by default in AWS. During failover, the interface from the primary node gets attached to the secondary node (the new primary node) and gets detached from the original primary node (the new secondary node).

13. Log on to the primary node with user name `nsroot` and the instance ID as the password. From the GUI, go to **System > High Availability**. The NetScaler VPX HA pair appears.

Next, configure the HA pairing on both the instances. Configure the instance with three ENIs before configuring HA on the instance with one ENI). Use the add HA node command, from within the VPX CLI, or from the GUI.

add HA node <private IP of the first instance>

add HA node <private IP of the second instance>

After you run the "add HA node" commands, the two nodes form an HA pair, and configuration information is synchronized between the two VPX instances.

Configuring SR-IOV on a high availability setup

Support for SR-IOV interfaces in a high availability setup is available from NetScaler release 12.0 57.19 onwards. For more information about how to configure SR-IOV, see Configuring NetScaler VPX instances to Use SR-IOV Network Interface.

Upgrade a NetScaler VPX instance on AWS

September 21, 2020

You can upgrade the EC2 instance type, throughput, software edition, and the system software of a NetScaler VPX running on AWS. For certain types of upgrades, Citrix recommends using the High Availability Configuration method to minimize downtime.

> **Note:**
>
> - NetScaler software release 10.1.e-124.1308.e or later for a NetScaler VPX AMI (including both utility license and customer license) does not support the M1 and M2 instance families.
> - Because of changes in VPX instance support, downgrading from 10.1.e-124 or a later release to 10.1.123.x or an earlier release is not supported.
> - Most of the upgrades do not require the launch of a new AMI, and the upgrade can be done on the current NetScaler AMI instance. If you do want to upgrade to a new NetScaler AMI instance, use the high availability configuration method.

Change the EC2 instance type of a NetScaler VPX instance on AWS

If your NetScaler VPX instances are running release 10.1.e-124.1308.e or later, you can change the EC2 instance type from the AWS console as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

You can also use the above procedure to change the EC2 instance type for a release, earlier than 10.1.e-124.1308.e, unless you want to change the instance type to M3. In that case, you must first follow the standard NetScaler upgrade procedure, at , to upgrade the NetScaler software to 10.1.e-124 or a later release, and then follow the above steps.

Upgrade the throughput or software edition of a NetScaler VPX instance on AWS

To upgrade the software edition (for example, to upgrade from Standard to Platinum edition) or throughput (for example, to upgrade from 200 mbps to 1000mbps), the method depends on the instance's license.

Using a customer license (Bring-Your-Own-License)

If you are using a customer license, you can purchase and download the new license from the Citrix website, and then install the license on the VPX instance. For more information about downloading and installing a license from the Citrix website, see the VPX Licensing Guide.

Using a utility license (Utility license with hourly fee)

AWS does not support direct upgrades for fee-based instances. To upgrade the software edition or throughput of a fee based NetScaler VPX instance, launch a new AMI with the desired license and capacity and migrate the older instance configuration to the new instance. This can be achieved by using a NetScaler high availability configuration as described in Upgrade to a new NetScaler AMI instance by using a NetScaler high availability configuration subsection in this page.

Upgrade the system software of a NetScaler VPX instance on AWS

If you need to upgrade a VPX instance running 10.1.e-124.1308.e or a later release, follow the procedure at Upgrade and downgrade a NetScaler appliance.

If you need to upgrade a VPX instance running a release older than 10.1.e-124.1308.e to 10.1.e-124.1308.e or a later release, first upgrade the system software, and then change the instance type to M3 as follows:

1. Stop the VPX instance.
2. Change the EC2 instance type from the AWS console.
3. Start the instance.

Upgrade to a new NetScaler AMI instance by using a NetScaler high availability configuration

To use the high availability method of upgrading to a new NetScaler AMI instance, perform the following tasks:

- Create a new instance with the desired EC2 instance type, software edition, throughput, or software release from the AWS marketplace.
- Configure high availability between the old instance (to be upgraded) and the new instance. After high availability is configured between the old and the new instance, configuration from the old instance is synchronized to the new instance.
- Force an HA failover from the old instance to the new instance. As a result, the new instance becomes primary and starts receiving traffic.
- Stop, and reconfigure or remove the old instance from AWS.

Prerequisites and points to consider

- Ensure you understand how high availability works between two NetScaler VPX instances on AWS. For more information about high availability configuration between two NetScaler VPX instances on AWS, see Deploy a high availability pair on AWS.

- You must create the new instance in the same availability zone as the old instance, having the exact same security group and subnet.
- High availability setup requires access and secret keys associated with the user's AWS Identity and Access Management (IAM) account for both instances. If the correct key information is not used when creating VPX instances, the HA setup fails. For more information about creating an IAM account for a VPX instance, see Prerequisites.
  - You must use the EC2 console to create the new instance. You cannot use the AWS 1-click launch, because it does not accept the access and secret keys as the input.
  - The new instance should have only one ENI interface.

To upgrade a NetScaler VPX Instance by using a high availability configuration, follow these steps:

1. Configure high availability between the old and the new instance. To configure high availability between two NetScaler VPX instances, at the command prompt of each instance, type:

   - add ha node <nodeID> <IPaddress of the node to be added>
   - save config

   Example:

   At the command prompt of the old instance, type:

   ```
   add ha node 30 192.0.2.30
   Done
   <!--NeedCopy-->
   ```

   At the command prompt of the new instance, type:

   ```
   add ha node 10 192.0.2.10
   Done
   <!--NeedCopy-->
   ```

   Note the following:

   - In the HA setup, the old instance is the primary node and the new instance is the secondary node.
   - The NSIP IP address is not copied from the old instance to the new instance. Therefore, after the upgrade, your new instance has a different management IP address from the previous one.
   - The nsroot account password of the new instance is set to that of the old instance after HA synchronization.

   For more information about high availability configuration between two NetScaler VPX instances on AWS, see Deploy a high availability pair on AWS.

2. Force an HA failover. To force a failover in a high availability configuration, at the command prompt of either of the instances, type:

```
force HA failover
<!--NeedCopy-->
```

As the result of forcing a failover, the ENIs of the old instance are migrated to the new instance and traffic flows through the new instance (the new primary node). The old instance (the new secondary node) restarts.
If the following warning message appears, type N to abort the operation:

```
[WARNING]:Force Failover may cause configuration loss, peer health
    not optimum. Reason(s):
HA version mismatch
HA heartbeats not seen on some interfaces
Please confirm whether you want force-failover (Y/N)?
<!--NeedCopy-->
```

The warning message appears because the system software of the two VPX instances is not HA compatible. As a result, the configuration of the old instance cannot be automatically synced to the new instance during a forced failover.

Following is the workaround for this issue:

1. At the NetScaler shell prompt of the old instance, type the following command to create a backup of the configuration file (ns.conf):
   copy /nsconfig/ns.conf to /nsconfig/ns.conf.bkp
2. Remove the following line from the backup configuration file (ns.conf.bkp):

   ```
   set ns config -IPAddress \<IP\> -netmask \<MASK\>
   ```

   For example, set ns config -IPAddress 192.0.2.10 -netmask 255.255.255.0
3. Copy the old instance's backup configuration file (ns.conf.bkp) to the /nsconfig directory of the new instance.
4. At the NetScaler shell prompt of the new instance, type the following command to load the old instance's configuration file (ns.conf.bkp) on the new instance:

   ```
   batch -f /nsconfig/ns.conf.bkp
   ```

5. Save the configuration on the new instance.

   ```
   Save conifg
   ```

6. At the command prompt of either of the nodes, type the following command to force a failover. Then type Y, for the warning message to confirm the force failover operation:

`force ha failover`

Example:

```
force ha failover

[WARNING]:Force Failover may cause configuration loss, peer health
     not optimum.
Reason(s):
HA version mismatch
HA heartbeats not seen on some interfaces
Please confirm whether you want force-failover (Y/N)? Y
<!--NeedCopy-->
```

7. Remove the HA configuration, so that the two instances are no longer in an HA configuration. First remove the HA configuration from the secondary node and then remove the HA configuration from the primary node.

To remove an HA configuration between two NetScaler VPX instances, at the command prompt of each instance, type:

```
         > remove ha node \<nodeID\>
         >  save config
<!--NeedCopy-->
```

For more information about high availability configuration between two VPX instances on AWS, see Deploy a high availability pair on AWS.

Example:

At the command prompt of the old instance (new secondary node), type:

```
         > remove ha node 30
           Done
         > save config
           Done
<!--NeedCopy-->
```

At the command prompt of the new instance (new primary node), type:

```
         > remove ha node 10
           Done
         > save config
           Done
<!--NeedCopy-->
```

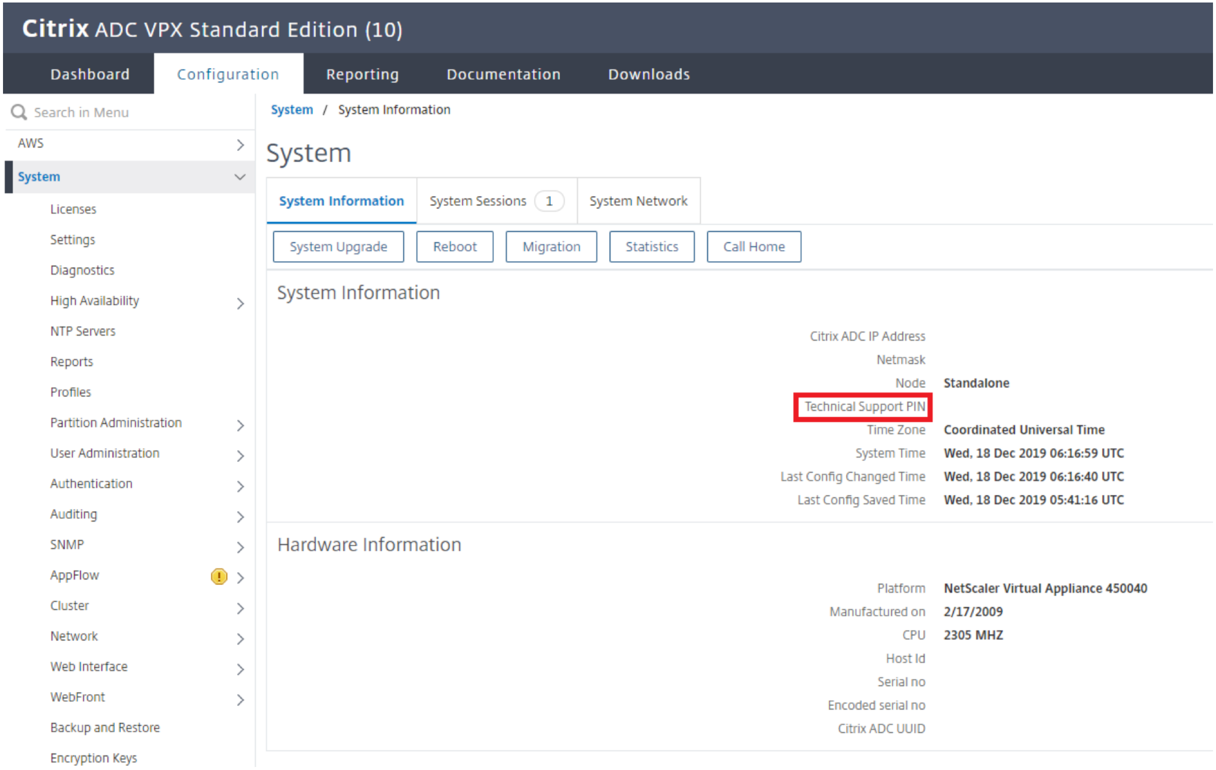
Troubleshoot a VPX instance on AWS

September 21, 2020

Amazon does not provide console access to a NetScaler VPX instance. To troubleshoot, you have to use the AWS GUI to view the activity log. You can debug only if the network is connected. To view an instance's system log, right-click the instance and select system log.

Citrix provides support for fee based NetScaler VPX instances (utility license with hourly fee) on AWS. To file a support case, find your AWS account number and support PIN code, and call Citrix support. You will also be asked for your name and email address. To find the support PIN, log on to the VPX GUI and navigate to the System page.

Here is an example of a system page showing the support PIN.


Citrix ADC VPX Standard Edition (10)
Dashboard
Configuration
Reporting
Documentation
Downloads
Search in Menu
AWS
System
Licenses
Settings
Diagnostics
High Availability
NTP Servers
Reports
Profiles
Partition Administration
User Administration
Authentication
Auditing
SNMP
AppFlow
Cluster
Network
Web Interface
WebFront
Backup and Restore
Encryption Keys
System / System Information
System
System Information
System Sessions 1
System Network
System Upgrade
Reboot
Migration
Statistics
Call Home
System Information
Citrix ADC IP Address
Netmask
Technical Support PIN
Hardware Information
Host Id
Serial no
Encoded serial no
Citrix ADC UUID

Node Standalone
Time Zone Coordinated Universal Time
System Time Wed, 18 Dec 2019 06:16:59 UTC
Last Config Changed Time Wed, 18 Dec 2019 06:16:40 UTC
Last Config Saved Time Wed, 18 Dec 2019 05:41:16 UTC

Platform NetScaler Virtual Appliance 450040
Manufactured on 2/17/2009
CPU 2305 MHZ

Deploying Citrix NetScaler VPX on Microsoft Azure

April 16, 2021

Jun 29, 2017

Microsoft Azure Resource Manager (ARM) is a management framework that allows administrators to deploy, manage, and monitor Azure resources. Azure Resource Manager can handle these tasks as a group, rather than individually, in a single operation.

The NetScaler VPX virtual appliance is available as an image in the Microsoft Azure Marketplace. When you deploy NetScaler VPX on Microsoft Azure Resource Manager (ARM), you can use the Azure cloud computing capabilities and use NetScaler load balancing and traffic management features for your business needs. You can deploy NetScaler VPX instances on Azure Resource Manager either as standalone instances or as high availability pairs in active-active or active-standby modes.

> **Note**
>
> This document describes how NetScaler VPX works when deployed with Azure Resource Manager (ARM). For information about NetScaler VPX deployment and architecture in Azure Cloud Services, see the details provided in Deploying NetScaler VPX on Azure 10.5 release.

This document assumes that you are familiar with Azure terminology and network details. For information about Microsoft Azure services, see Microsoft Azure Documentation Center.

This document also assumes that you have basic knowledge of a NetScaler appliance. For detailed information about NetScaler appliances, see:

- NetScaler
- NetScaler Gateway

This document provides information about:

- Network Architecture
- How NetScaler VPX Works on Azure
- Traffic Flow through Port Address Translation
- Traffic Flow through Network Address Translation
- Port Usage Guidelines
- Limitations

For information about NetScaler Gateway configuration for Citrix XenApp and XenDesktop in Azure cloud, see https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/netscaler-vpx-deployment-with-xendesktop-and-xenapp-on-microsoft-azure.pdf.

> **Note**
>
> The Citrix XenApp and XenDesktop NetScaler Gateway configuration is based on the Azure Service Management mode and not on the Azure Resource Management mode.

Network Architecture

In ARM, a NetScaler VPX virtual machine (VM) resides in a virtual network. A virtual NIC is created on each NetScaler VM. The network security group (NSG) configured in the virtual network is bound to the NIC, and together they control the traffic flowing into the VM and out of the VM.

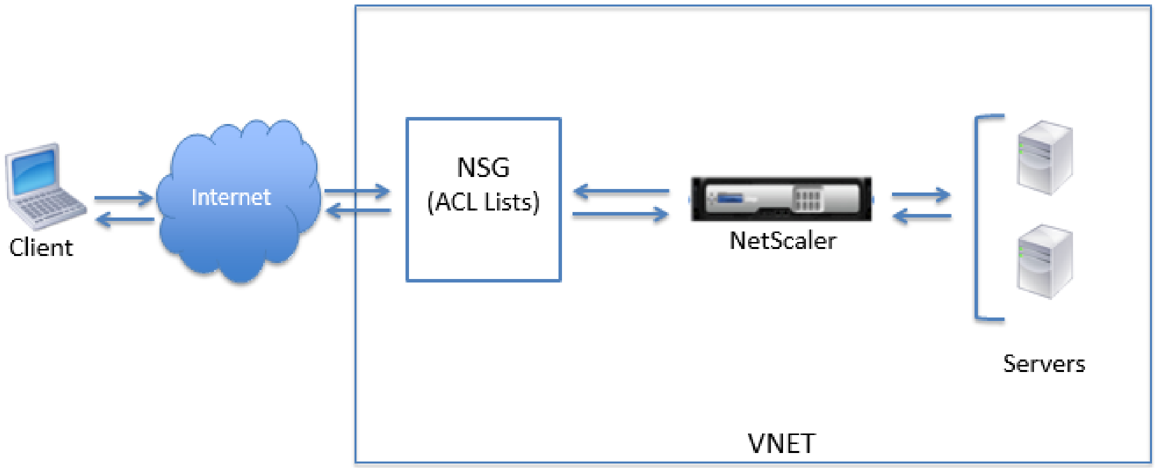
The NSG forwards the requests to the NetScaler VPX instance, and the VPX instance sends them to the servers. The response from a server follows the same path in reverse. The NSG can be configured to control a single VPX VM, or, with subnets and virtual networks, can control traffic in multiple VPX VM deployments.

The NIC contains network configuration details such as the virtual network, subnets, internal IP address, and Public IP address.

While on ARM, it is good to know the following IP addresses used to access the VMs:

- Public IP (PIP) address is the Internet-facing IP address configured directly on the virtual NIC of the NetScaler VM. This allows you to directly access a VM from the external network without the need to configure inbound and outbound rules on the NSG.
- NetScaler IP (NSIP) address is internal IP address configured on the VM. It is non-routable.
- Virtual IP address (VIP) is configured by using the NSIP and a port number. Clients access NetScaler services through the PIP address, and when the request reaches the NIC of the NetScaler VPX VM or the Azure load balancer, the VIP gets translated to internal IP (NSIP) and internal port number.
- Internal IP address is the private internal IP address of the VM from the virtual network's address space pool. This IP address cannot be reached from the external network. This IP address is by default dynamic unless you set it to static. Traffic from the internet is routed to this address according to the rules created on the NSG. The NSG works with the NIC to selectively send the right type of traffic to the right port on the NIC, which depends on the services configured on the VM.

The following figure shows how traffic flows from a client to a server through a NetScaler VPX instance provisioned in ARM.



How NetScaler VPX Works on Azure

In an on-premises deployment, a NetScaler VPX instance requires at least three IP addresses:

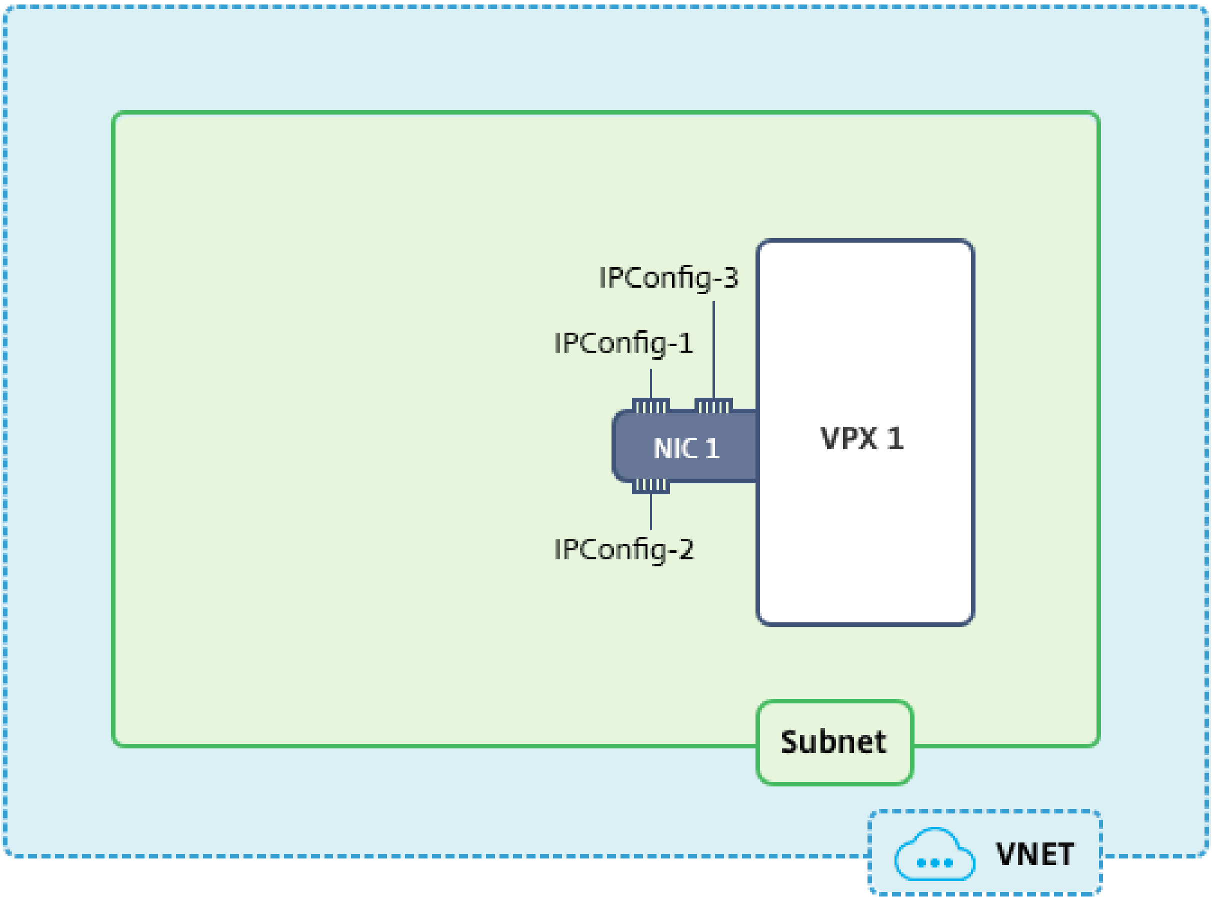
- Management IP address, called the NetScaler IP (NSIP) address
- Subnet IP (SNIP) address for communicating with the server farm
- Virtual server IP (VIP) address for accepting client requests

In an Azure deployment, you can provision a NetScaler VPX instance in Azure in two ways:

- With a multi-NIC, multi-IP architecture
- With a single-IP architecture

> **Note**
>
> VPX virtual appliances can be deployed on any instance type that has two or more Intel VT-X cores and more than 2 GB memory. For more information on system requirements, see Citrix ADC VPX data sheet. Currently, NetScaler VPX instance supports only the Intel processors.

The following image illustrates how multiple IP addresses are used to perform the functions of NSIP, SNIP, and VIP, with a single NIC in a standalone deployment. According to your requirement, you can configure multiple NICs with different IP addresses.

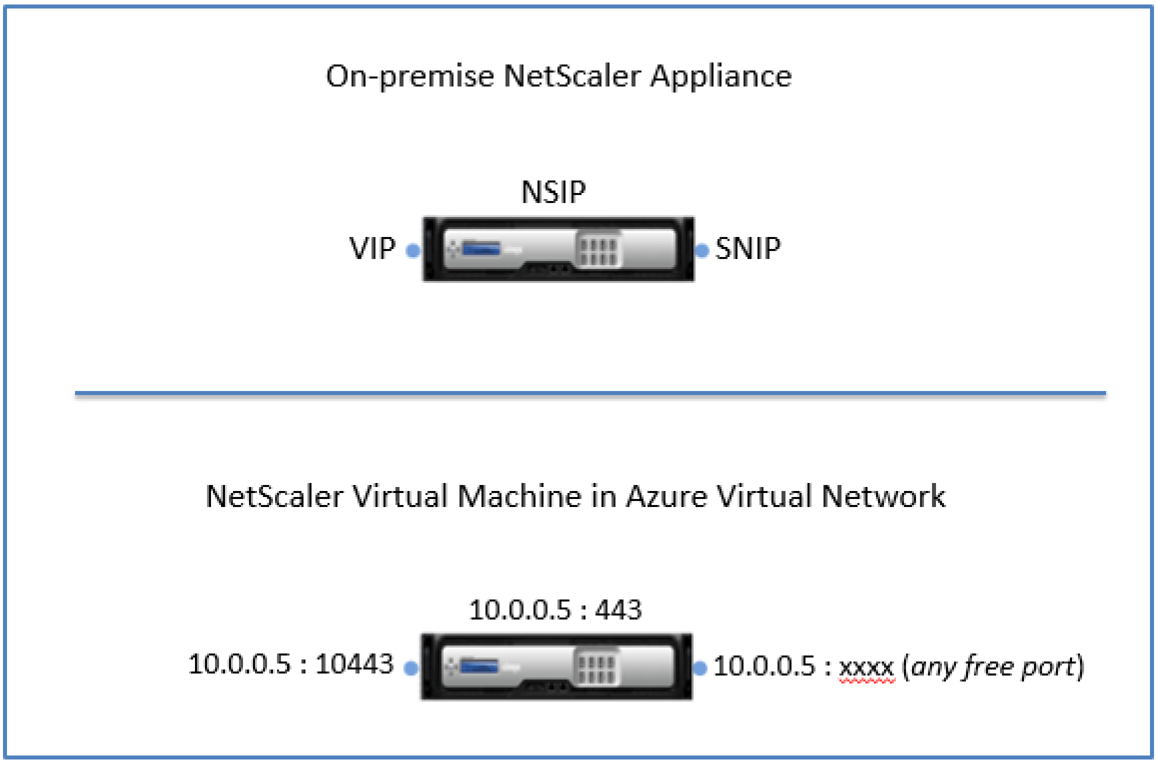


For more information about NetScaler multi-NIC, multi-IP deployment on Azure, see the following links:

- Configure multiple IP addresses for a NetScaler standalone instance
- Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands

In single IP mode the three IP functions of a NetScaler appliance are multiplexed onto one IP address. This single IP address is assigned to an instance during provisioning through DHCP, and it is a private (internal) address and it uses different port numbers to function as the NSIP, SNIP, and VIP.

The following image illustrates how a single IP address is used to perform the functions of NSIP, SNIP, and VIP.


On-premise NetScaler Appliance
NSIP
VIP
SNIP
NetScaler Virtual Machine in Azure Virtual Network
10.0.0.5 : 443
10.0.0.5 : 10443
10.0.0.5 : xxxx (*any free port*)

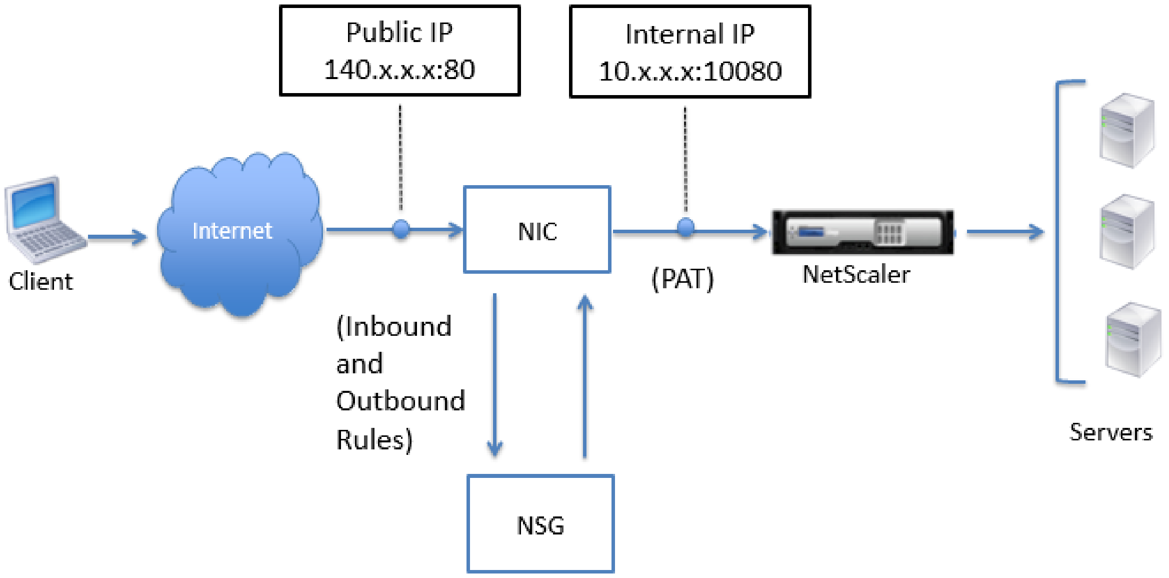
> **Note**
>
> The single IP mode is available only in Azure deployments. This mode is not available for a NetScaler VPX instance on your premises, on AWS, or in other type of deployment.

Traffic Flow through Port Address Translation

In an Azure deployment, when you provision the NetScaler VPX instance as a virtual machine (VM), Azure assigns a public IP (PIP) address and an internal IP address (non-routable) to the NetScaler virtual machine. Inbound and Outbound rules are defined on the NSG for the NetScaler instance, along with a public port and a private port for each rule defined. The NetScaler instance listens on the internal IP address and private port.

Any external request is received on the NetScaler VPX VM's virtual NIC. The NIC is bound to the NSG, which specifies the private IP and private port combination into which to translate the request's destination address and port (the public IP address and port). ARM performs port address translation (PAT) to map the Public IP address and port to the internal IP address and private port of the NetScaler virtual machine, and forwards the traffic to the virtual machine.

The following figure shows how Azure performs port address translation to direct traffic to the NetScaler internal IP address and private port.



In this example, the Public IP address assigned to the VM is 140.x.x.x, and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the NetScaler virtual machine listens. The client request is received on the Public IP address 140.x.x.x at port 80. Azure performs port address translation to map this address and port to internal IP address 10.x.x.x on private port 10080 and forwards the client request.

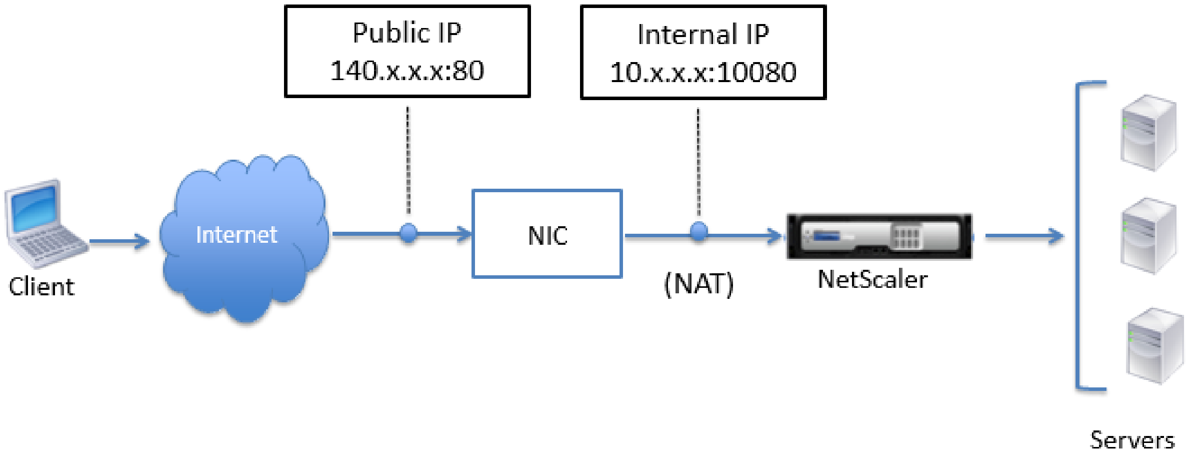
For information about port usage guidelines while, see Port Usage Guidelines.

For information about NSG and access control lists, see What is a Network Security Group?

Traffic Flow Through Network Address Translation

You can also request a Public IP (PIP) address for your NetScaler virtual machine (instance level). If you use this direct PIP at the VM level, you need not define inbound and outbound rules to intercept the network traffic. The incoming request from the Internet is received on the VM directly. Azure performs network address translation (NAT) and forwards the traffic to the internal IP address of the NetScaler instance.

The following figure shows how Azure performs network address translation to map the NetScaler internal IP address.



In this example, the Public IP assigned to the NSG is 140.x.x.x and the internal IP address is 10.x.x.x. When the inbound and outbound rules are defined, public HTTP port 80 is defined as the port on which the client requests are received, and a corresponding private port, 10080, is defined as the port on which the NetScaler virtual machine listens. The client request is received on the Public IP address (140.x.x.x). Azure performs network address translation to map the PIP to the internal IP address 10.x.x.x on port 10080, and forwards the client request.

> **Note**
>
> NetScaler VPX VMs in high availability are controlled by external or internal load balancers that have inbound rules defined on them to control the load balancing traffic. The external traffic is first intercepted by these load balancers and the traffic is diverted according to the load balancing rules configured, which has back-end pools, NAT rules, and health probes defined on the load balancers.

Port Usage Guidelines

You can configure more inbound and outbound rules n NSG while creating the NetScaler virtual machine or after the virtual machine is provisioned. Each inbound and outbound rule is associated with a public port and a private port.

Before configuring NSG rules, note the following guidelines regarding the port numbers you can use:

1. The following ports are reserved by the NetScaler virtual machine. You cannot define these as

private ports when using the Public IP address for requests from the Internet.
Ports 21, 22, 80, 443, 8080, 67, 161, 179, 500, 520, 3003, 3008, 3009, 3010, 3011, 4001, 5061, 9000, 7000.
However, if you want Internet-facing services such as the VIP to use a standard port (for example, port 443) you have to create port mapping by using the NSG. The standard port is then mapped to a different port that is configured on the NetScaler for this VIP service.
For example, a VIP service might be running on port 8443 on the NetScaler instance but be mapped to public port 443. So, when the user accesses port 443 through the Public IP, the request is directed to private port 8443. However, if you want internet-facing services such as the VIP to use a standard port (for example, port 443) you have to create port mapping by using the NSG. The standard port is then mapped to a different port that is configured on the NetScaler for this VIP service.
Note: If you access the NetScaler instance in Network Access mode (Full SSL VPN), you need to map the private port 8443 to the public port 8443.

2. Public IP address does not support protocols in which port mapping is opened dynamically, such as passive FTP or ALG.
3. High availability does not work for traffic that uses a public IP address (PIP) associated with a VPX instance, instead of a PIP configured on the Azure load balancer. For more information about configuring NetScaler VPX HA in ARM, see Configuring Multiple Azure NICs and IP Addresses for NetScaler VPX Instances in HA Mode.
4. In a NetScaler Gateway deployment, you need not configure a SNIP address, because the NSIP can be used as a SNIP when no SNIP is configured.
   You must configure the VIP address by using the NSIP address and some nonstandard port number. For call-back configuration on the back end server, the VIP port number has to be specified along with the VIP URL (for example, url:port).

> **Note**
>
> In Azure Resource Manager, a NetScaler VPX instance is associated with two IP addresses - a public IP address (PIP) and an internal IP address. While the external traffic connects to the PIP, the internal IP address or the NSIP is non-routable. To configure VIP in VPX, use the internal IP address (NSIP) and any of the free ports available. Do not use the PIP to configure VIP.
>
> Example: If NSIP of a NetScaler VPX is 10.1.0.3 and an available free port is 10022, then you can configure VIP by providing the 10.1.0.3:10022 (NSIP address+port) combination.

Limitations

Running the NetScaler VPX load balancing solution on ARM imposes the following limitations:

1. The Azure architecture does not accommodate support for the following NetScaler features:

Clustering

IPv6

Gratuitous ARP (GARP)

L2 Mode

Tagged VLAN

Dynamic Routing

Virtual MAC

USIP

CloudBridge Connector

- If you expect that you might have to shut down and temporarily deallocate the NetScaler VPX virtual machine at any time, assign a static Internal IP address while creating the virtual machine. If you do not assign a static internal IP address, Azure might assign the virtual machine a different IP address each time it restarts, and the virtual machine might become inaccessible.
- Citrix ADC VPX instances lower than D16sv3 that are configured for ICA Proxy can experience a high degree of latency. Hence, Citrix recommends you to use the instance size of D16sv3 or D32sv3.
- In an Azure deployment, only the following NetScaler VPX models are supported: VPX 10, VPX 25, VPX 200, VPX 1000, and VPX 3000. For for information, see the NetScaler VPX Data Sheet.
  If you use a NetScaler VPX instance with a model number higher than VPX 3000, the network throughput might not be the same as specified by the instance's license. However, other features, such as SSL throughput and SSL transactions per second, might improve.
- The "deployment ID" that is generated by Azure during virtual machine provisioning is not visible to the user in ARM. You cannot use the deployment ID to deploy NetScaler VPX appliance on ARM.
- Active-passive or active-standy HA mode is not supported for VPX configured with multiple NICs and multiple IP addresses.
- The NetScaler VPX appliance supports 5 Mb/s throughput and standard edition features when it's initialized.
- For a XenApp and XenDesktop deployment, a VPN virtual server on a NetScaler appliance can be configured in the following modes:
- Basic mode, where the ICAOnly VPN virtual server parameter is set to ON. The Basic mode works fully on an unlicensed NetScaler VPX instance.
- SmartAccess mode, where the ICAOnly VPN virtual server parameter is set to OFF. The SmartAccess mode works for only 5 AAA session users on an unlicensed NetScaler VPX instance.

> **Note**
>
> To configure the SmartControl feature, you must apply a platinum license to the NetScaler VPX instance.

Configure a NetScaler VPX standalone instance

September 21, 2020

You can provision a single NetScaler VPX instance in Azure Resource Manager (ARM) portal in a standalone mode by creating the virtual machine and configuring other resources.

Before you begin

Make sure that you have the following:

- A Microsoft Azure user account
- Access to Microsoft Azure Resource Manager
- Microsoft Azure SDK
- Microsoft Azure PowerShell

On the Microsoft Azure Portal page, log on to the Azure Resource Manager portal by providing your user name and password.

> **Note**
>
> In ARM portal, clicking an option in one pane opens a new pane to the right. Navigate from one pane to another to configure your device.

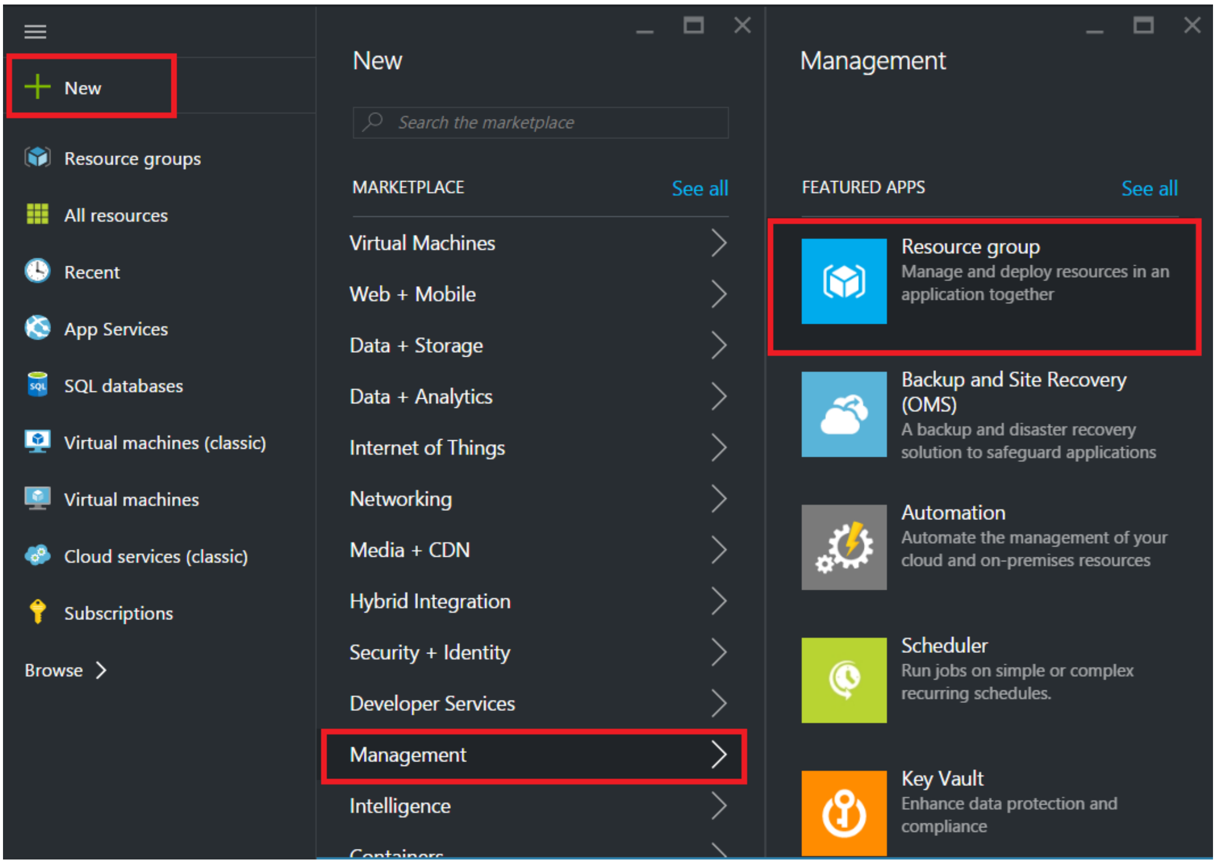
Summary of configuration steps

1. Configure a resource group
2. Configure a network security group
3. Configure virtual network and its subnets
4. Configure a storage account
5. Configure an availability set
6. Configure a NetScaler VPX instance

Configure a resource group

Create a new resource group that is a container for all your resources. Use the resource group to deploy, manage, and monitor your resources as a group.

1. Click **New** > **Management** > **Resource group**.
2. In the **Resource group** pane, enter the following details:
   - Resource group name
   - Resource group location
3. Click **Create**.

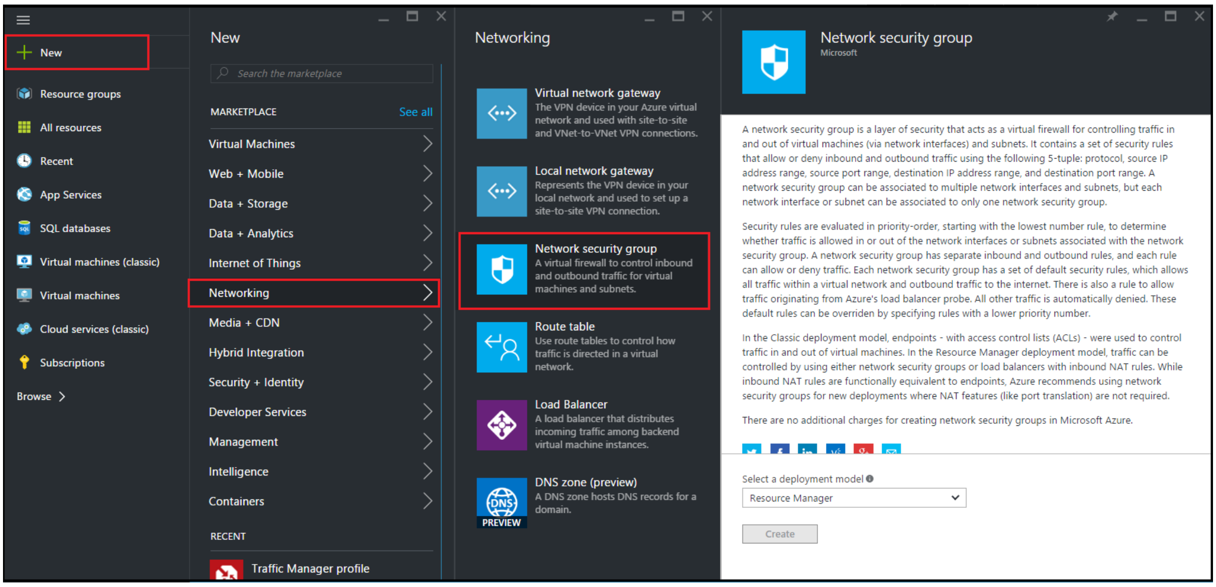

New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Browse >
New
Search the marketplace
MARKETPLACE
See all
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Management
FEATURED APPS
See all
Resource group
Manage and deploy resources in an application together
Backup and Site Recovery (OMS)
A backup and disaster recovery solution to safeguard applications
Automation
Automate the management of your cloud and on-premises resources
Scheduler
Run jobs on simple or complex recurring schedules.
Key Vault
Enhance data protection and compliance

Configure a network security group

Create a network security group (NSG) to assign inbound and outbound rules to control the incoming and outgoing traffic within the virtual network. NSG allows you to define security rules for a single virtual machine and also to define security rules for a virtual network subnet.

1. Click **New** > **Networking** > **Network security group**.
2. In the **Create network security group** pane, enter the following details, and then click **Create**.
   - Name - type a name for the security group
   - Resource group - select the resource group from the drop-down list

> **Note**
>
> Ensure that you have selected the correct location. The list of resources that appear in the drop-down list is different for different locations.


New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Browse >
New
Search the marketplace
MARKETPLACE
See all
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Containers
RECENT
Traffic Manager profile
Networking
Virtual network gateway
Local network gateway
Network security group
Route table
Load Balancer
DNS zone (preview)
PREVIEW
Network security group
Microsoft
Select a deployment model
Resource Manager
Create

Virtual network gateway: The VPN device in your Azure virtual network and used with site-to-site and VNet-to-VNet VPN connections.

Local network gateway: Represents the VPN device in your local network and used to set up a site-to-site VPN connection.

Network security group: A virtual firewall to control inbound and outbound traffic for virtual machines and subnets.

Route table: Use route tables to control how traffic is directed in a virtual network.

Load Balancer: A load balancer that distributes incoming traffic among backend virtual machine instances.

DNS zone (preview): A DNS zone hosts DNS records for a domain.

A network security group is a layer of security that acts as a virtual firewall for controlling traffic in and out of virtual machines (via network interfaces) and subnets. It contains a set of security rules that allow or deny inbound and outbound traffic using the following 5-tuple: protocol, source IP address range, source port range, destination IP address range, and destination port range. A network security group can be associated to multiple network interfaces and subnets, but each network interface or subnet can be associated to only one network security group.

Security rules are evaluated in priority-order, starting with the lowest number rule, to determine whether traffic is allowed in or out of the network interfaces or subnets associated with the network security group. A network security group has separate inbound and outbound rules, and each rule can allow or deny traffic. Each network security group has a set of default security rules, which allows all traffic within a virtual network and outbound traffic to the internet. There is also a rule to allow traffic originating from Azure's load balancer probe. All other traffic is automatically denied. These default rules can be overriden by specifying rules with a lower priority number.

In the Classic deployment model, endpoints - with access control lists (ACLs) - were used to control traffic in and out of virtual machines. In the Resource Manager deployment model, traffic can be controlled by using either network security groups or load balancers with inbound NAT rules. While inbound NAT rules are functionally equivalent to endpoints, Azure recommends using network security groups for new deployments where NAT features (like port translation) are not required.

There are no additional charges for creating network security groups in Microsoft Azure.

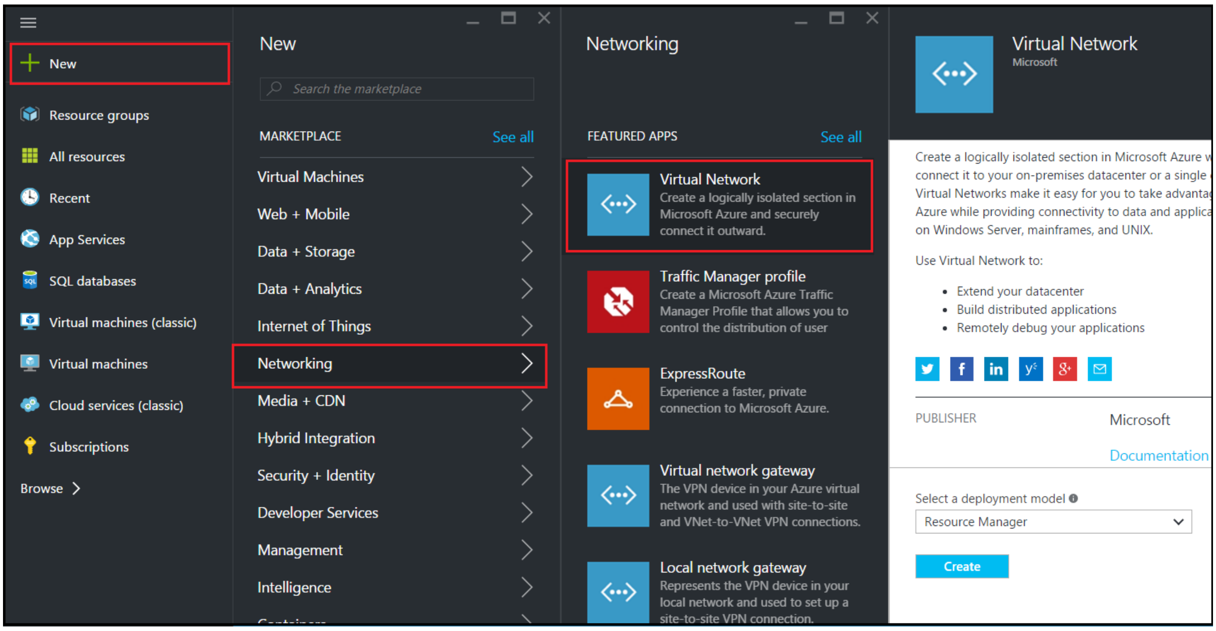
Select a deployment model: Resource Manager

Configure a virtual network and subnets

Virtual networks in ARM provide a layer of security and isolation to your services. VMs and services that are part of the same virtual network can access each other.

For these steps to create a virtual network and subnets.

1. Click **New** > **Networking** > **Virtual Network**.
2. In the **Virtual Network** pane, ensure the deployment mode is **Resource Manager** and click **Create**.


New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Browse >
New
Search the marketplace
MARKETPLACE
See all
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Networking
FEATURED APPS
See all
Virtual Network
Traffic Manager profile
ExpressRoute
Virtual network gateway
Local network gateway
Virtual Network
Microsoft
PUBLISHER
Microsoft
Documentation
Select a deployment model
Resource Manager
Create

Virtual Network: Create a logically isolated section in Microsoft Azure and securely connect it outward.

Traffic Manager profile: Create a Microsoft Azure Traffic Manager Profile that allows you to control the distribution of user

ExpressRoute: Experience a faster, private connection to Microsoft Azure.

Virtual network gateway: The VPN device in your Azure virtual network and used with site-to-site and VNet-to-VNet VPN connections.

Local network gateway: Represents the VPN device in your local network and used to set up a site-to-site VPN connection.

Create a logically isolated section in Microsoft Azure w
connect it to your on-premises datacenter or a single
Virtual Networks make it easy for you to take advanta
Azure while providing connectivity to data and applica
on Windows Server, mainframes, and UNIX.

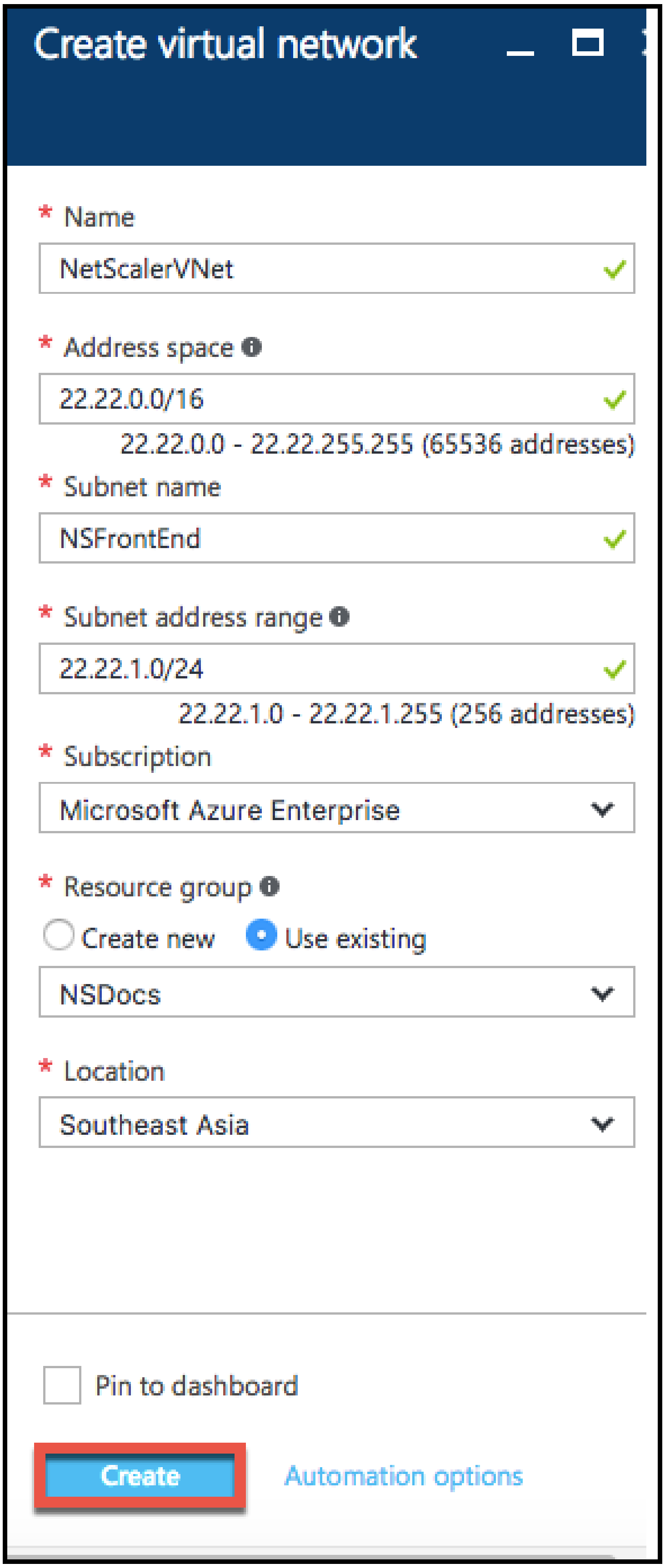
Use Virtual Network to:

- Extend your datacenter
- Build distributed applications
- Remotely debug your applications

Select a deployment model: Resource Manager

3. In the **Create virtual network** pane, enter the following values, and then click **Create**.
   - Name of the virtual network

- Address space - type the reserved IP address block for the virtual network
- Subnet - type the name of the first subnet (you will create the second subnet later in this step)
- Subnet address range - type the reserved IP address block of the subnet
- Resource group - select the resource group created earlier from the drop-down list



Create virtual network

* Name

NetScalerVNet ✓

* Address space

22.22.0.0/16 ✓

22.22.0.0 - 22.22.255.255 (65536 addresses)

* Subnet name

NSFrontEnd ✓

* Subnet address range

22.22.1.0/24 ✓

22.22.1.0 - 22.22.1.255 (256 addresses)

* Subscription

Microsoft Azure Enterprise

* Resource group

○ Create new ● Use existing

NSDocs

* Location

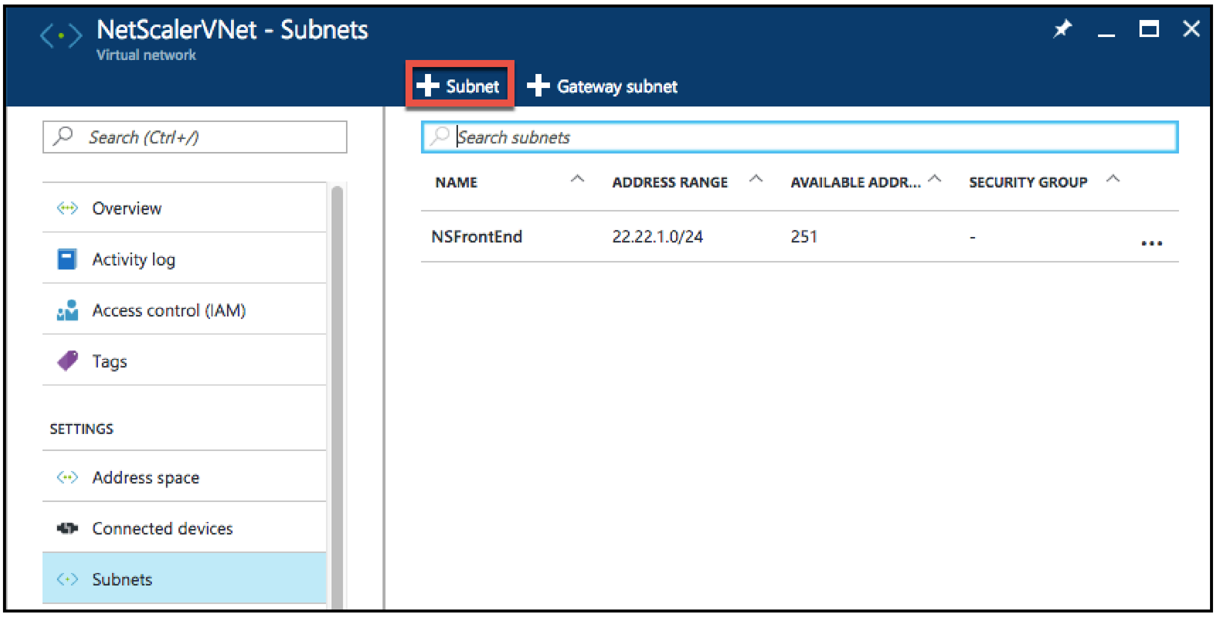
Southeast Asia

☐ Pin to dashboard

Create
Automation options

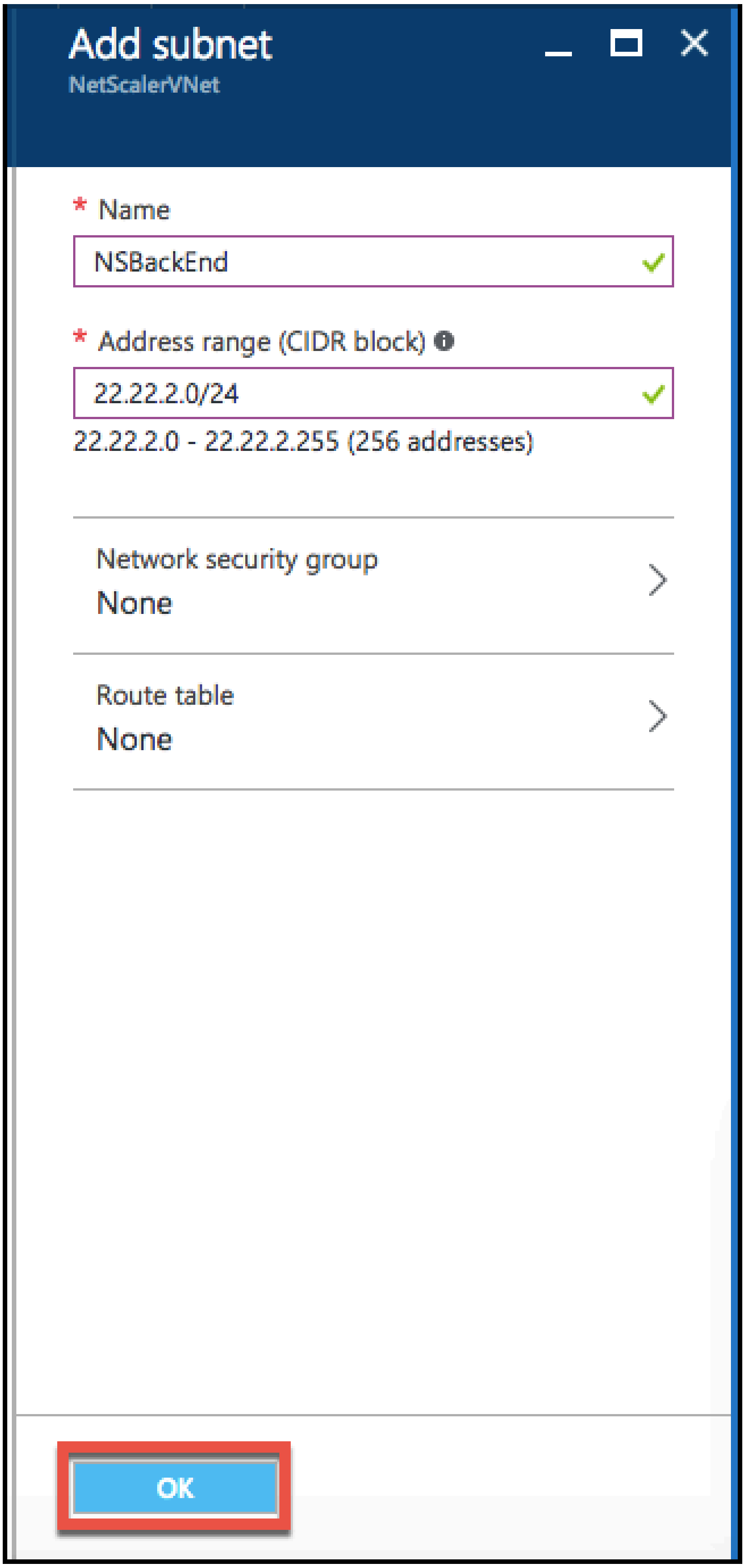
Configure the second subnet

1. Select the newly created virtual network from **All resources** pane and in the **Settings** pane, click **Subnets**.


NetScalerVNet - Subnets
Virtual network
+ Subnet
+ Gateway subnet
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
SETTINGS
Address space
Connected devices
Subnets
Search subnets

NAME	ADDRESS RANGE	AVAILABLE ADDR...	SECURITY GROUP
NSFrontEnd	22.22.1.0/24	251	-

2. Click **+Subnet** and create the second subnet by entering the following details.
   - Name of the second subnet
   - Address range - type the reserved IP address block of the second subnet
   - Network security group - select the NSG from the drop-down list
3. Click **Create**.



Add subnet
NetScalerVNet

* Name

NSBackEnd

* Address range (CIDR block)

22.22.2.0/24

22.22.2.0 - 22.22.2.255 (256 addresses)

Network security group

None

Route table

None

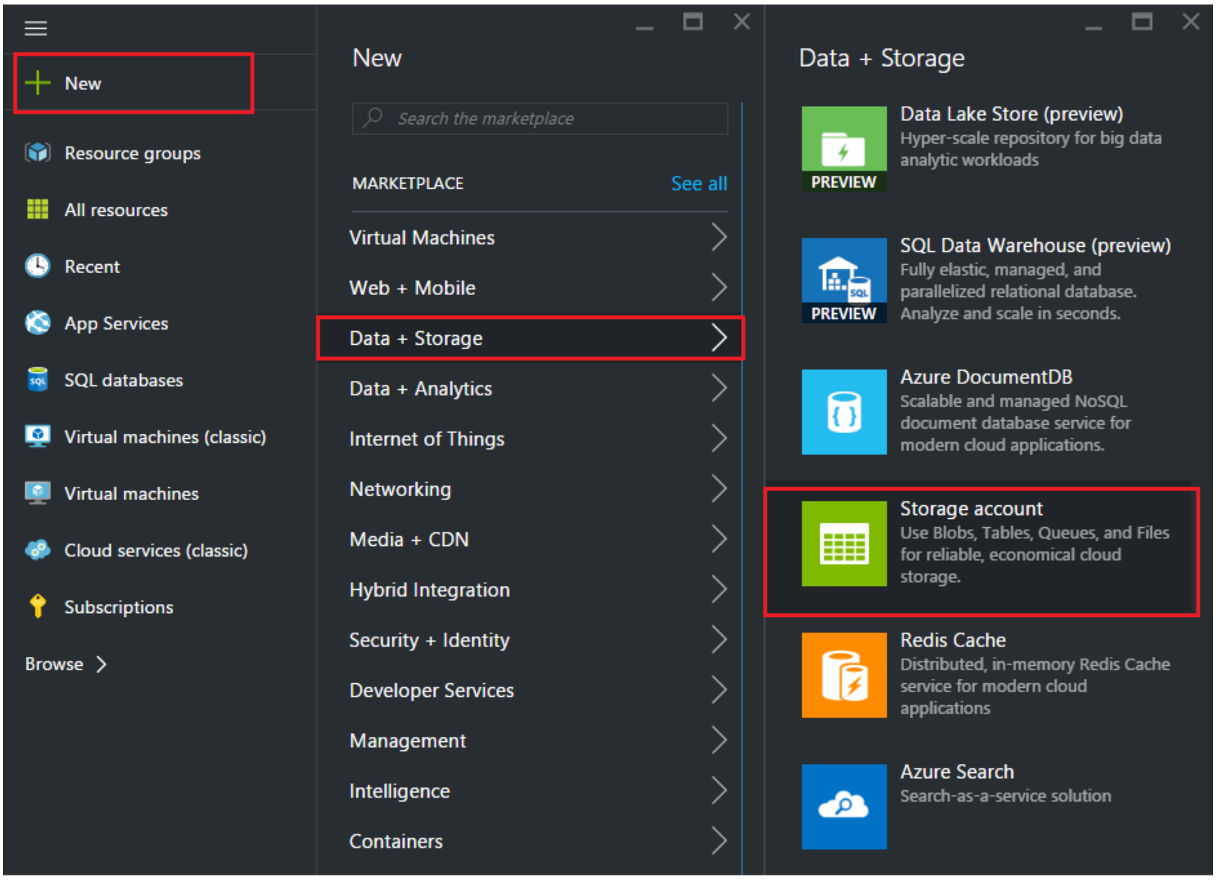
OK

Configure a storage account

The ARM IaaS infrastructure storage includes all services where we can store data in the form of blobs, tables, queues, and files. You can also create applications using these forms of storage data in ARM.

Create a storage account to store all your data.

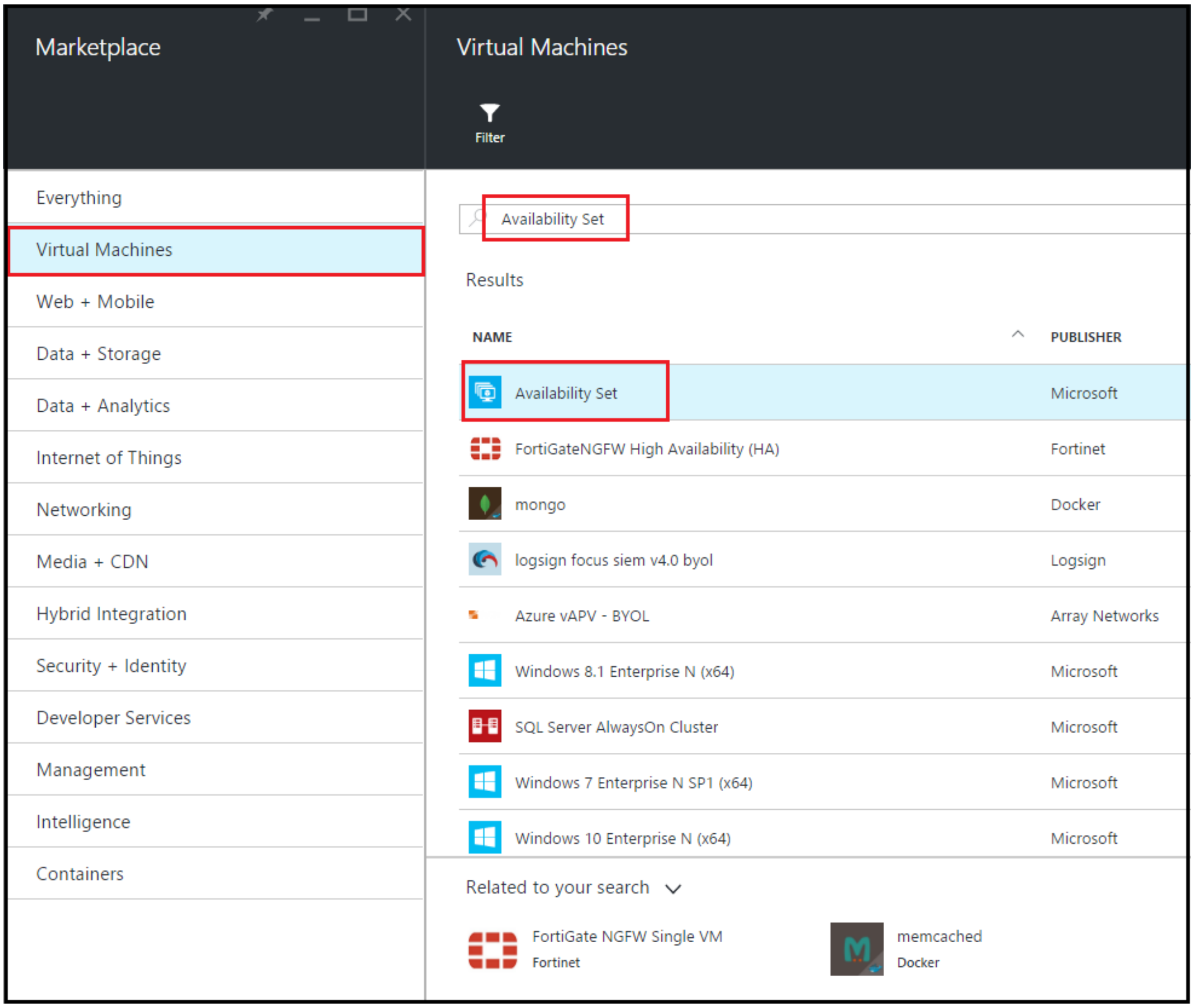
1. Click **+New** > **Data + Storage** > **Storage account**.
2. In the **Create storage account** pane, enter the following details:
    - Name of the account
    - Deployment mode - make sure to select **Resource Manager**
    - Account kind - select **General purpose** from the drop-down list
    - Replication - select **Locally redundant storage** from the drop-down list
    - Resource group - select the newly created resource group from the drop-down list
3. Click **Create**.


New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Browse
New
Search the marketplace
MARKETPLACE
See all
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Containers
Data + Storage
Data Lake Store (preview)
Hyper-scale repository for big data analytic workloads
PREVIEW
SQL Data Warehouse (preview)
Fully elastic, managed, and parallelized relational database. Analyze and scale in seconds.
PREVIEW
Azure DocumentDB
Scalable and managed NoSQL document database service for modern cloud applications.
Storage account
Use Blobs, Tables, Queues, and Files for reliable, economical cloud storage.
Redis Cache
Distributed, in-memory Redis Cache service for modern cloud applications
Azure Search
Search-as-a-service solution

Configure an availability set

An availability set guarantees that at least one VM is kept up and running in case of planned or unplanned maintenance. Two or more VMs under the same 'availability set' are placed on different fault domains to achieve redundant services.

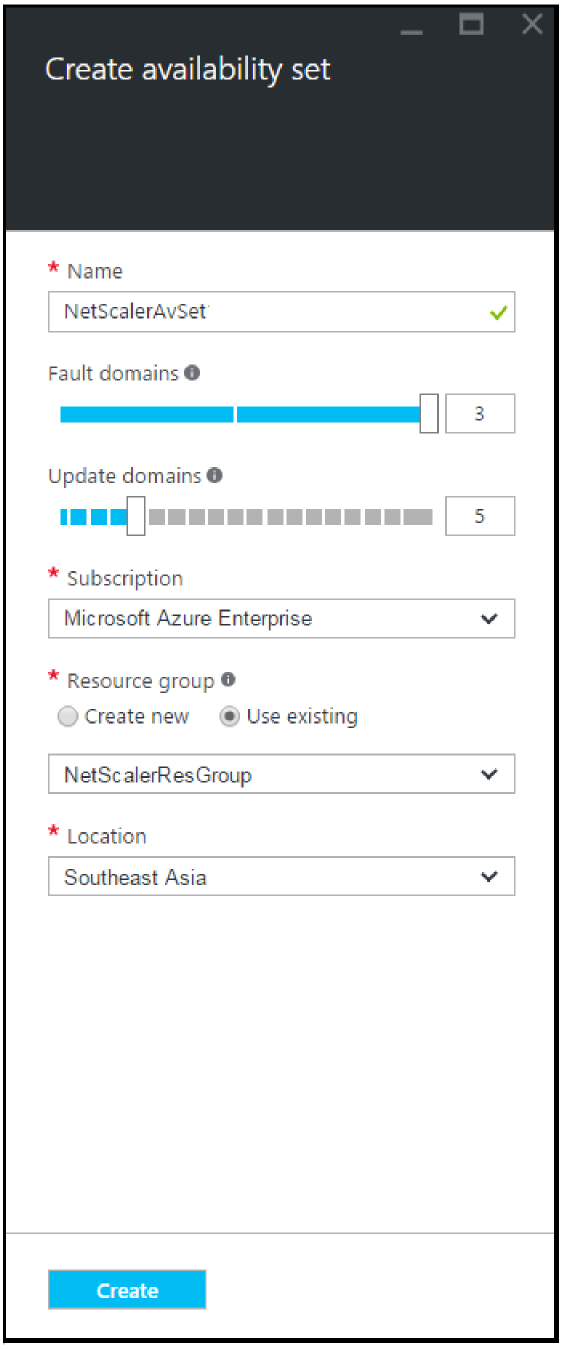
1. Click **+New**.
2. Click **See all** in the MARKETPLACE pane and click **Virtual Machines**.
3. Search for availability set, and then select **Availability set** entity from the list displayed.


Marketplace
Everything
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Containers
Virtual Machines
Filter
Results
Related to your search
FortiGate NGFW Single VM
Fortinet
memcached
Docker

Availability Set

NAME	PUBLISHER
Availability Set	Microsoft
FortiGateNGFW High Availability (HA)	Fortinet
mongo	Docker
logsign focus siem v4.0 byol	Logsign
Azure vAPV - BYOL	Array Networks
Windows 8.1 Enterprise N (x64)	Microsoft
SQL Server AlwaysOn Cluster	Microsoft
Windows 7 Enterprise N SP1 (x64)	Microsoft
Windows 10 Enterprise N (x64)	Microsoft

4. Click **Create,** and in the **Create availability set** pane, enter the following details:
   - Name of the set
   - Resource group - select the newly created resource group from the drop-down list
5. Click **Create**.



* Name: NetScalerAvSet

Fault domains: 3

Update domains: 5

* Subscription: Microsoft Azure Enterprise

* Resource group: ○ Create new ● Use existing

NetScalerResGroup

* Location: Southeast Asia

Configure a NetScaler VPX instance

Create an instance of NetScaler VPX in the virtual network. Obtain the NetScaler VPX image from the Azure marketplace, and then use the Azure Resource Manager portal to create a NetScaler VPX instance.

Before you begin creating the NetScaler VPX instance, make sure that you have created a virtual network with required subnets in which the instance will reside. You can create virtual networks during

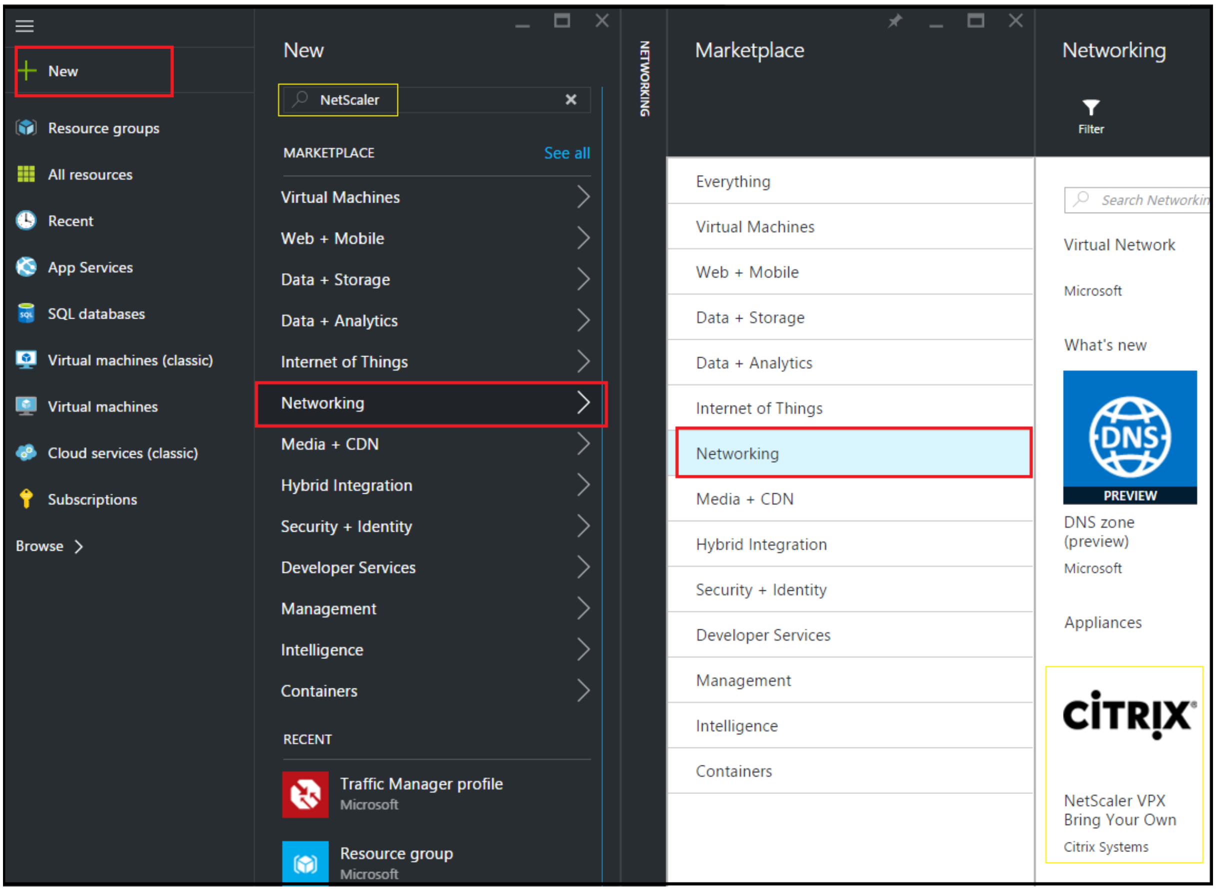
VM provisioning, but without the flexibility to create different subnets. For information about creating virtual networks, see http://azure.microsoft.com/en-us/documentation/articles/create-virtual-network/.

Optionally, configure DNS server and VPN connectivity that allows a virtual machine to access Internet resources.

> **Note**
>
> Citrix recommends that you create resource group, network security group, virtual network, and other entities before you provision the NetScaler VPX VM, so that the network information is available during provisioning.

1. Click **+New** > **Networking**.
2. Click **See All** and in the Networking pane, click **Citrix NetScaler VPX Bring Your Own License**.

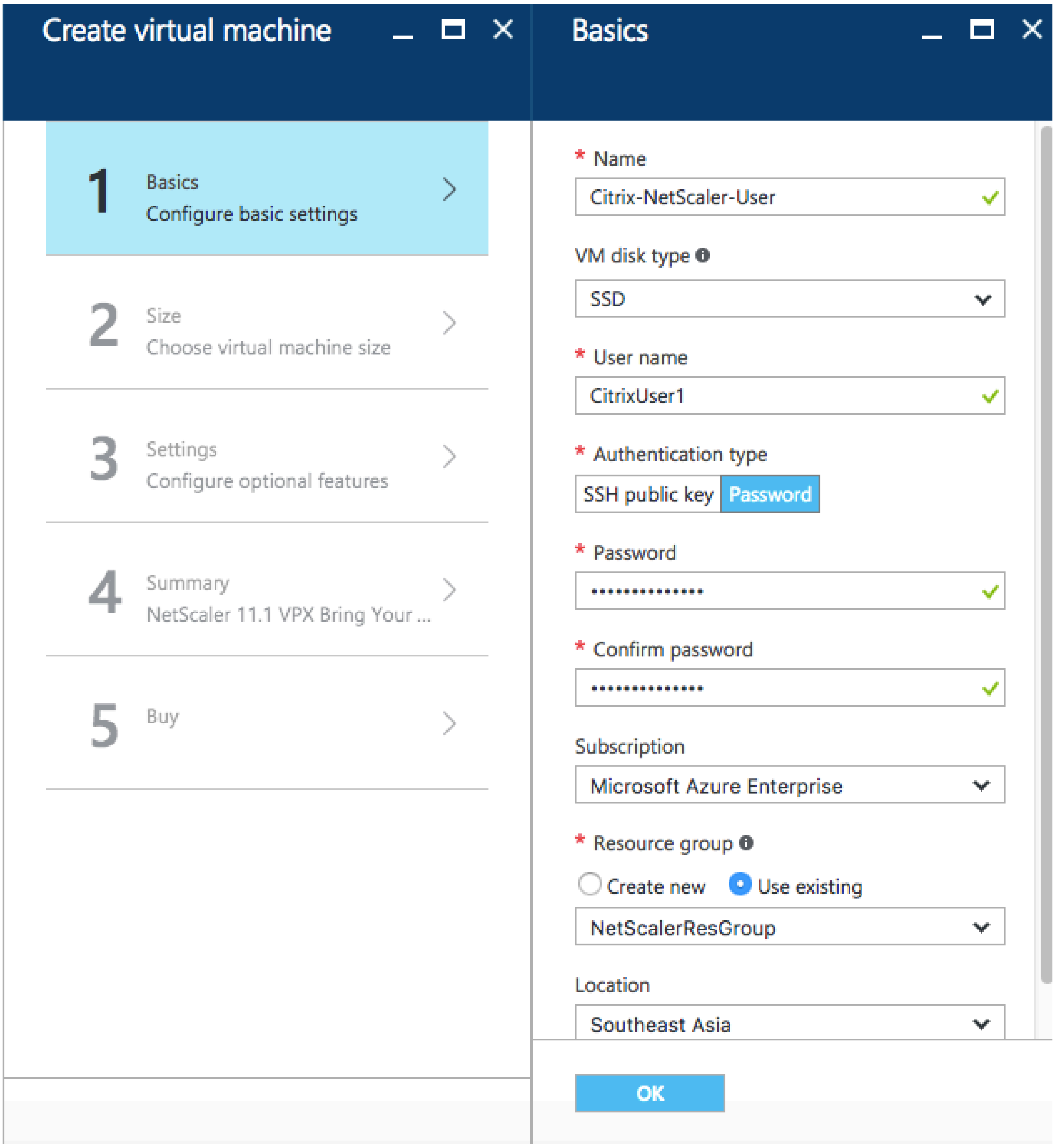
As a quick way to find any entity on ARM portal, you can also type the name of the entity in the Azure Marketplace search box and press <Enter>. Type NetScaler in the search box to find the Citrix NetScaler images.


New
Resource groups
All resources
Recent
App Services
SQL databases
Virtual machines (classic)
Virtual machines
Cloud services (classic)
Subscriptions
Browse >
New
NetScaler
MARKETPLACE
See all
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Containers
RECENT
Traffic Manager profile
Microsoft
Resource group
Microsoft
NETWORKING
Marketplace
Everything
Virtual Machines
Web + Mobile
Data + Storage
Data + Analytics
Internet of Things
Networking
Media + CDN
Hybrid Integration
Security + Identity
Developer Services
Management
Intelligence
Containers
Networking
Filter
Search Networkin
Virtual Network
Microsoft
What's new
DNS
PREVIEW
DNS zone (preview)
Microsoft
Appliances
CITRIX
NetScaler VPX Bring Your Own
Citrix Systems

> **Note**
>
> Ensure to select the latest image. Your Citrix NetScaler image might have the release number in

> the name.

3. On the **NetScaler VPX Bring Your Own License** page, from the drop-down list, select **Resource Manager** and click **Create**.



Create virtual machine

1 Basics
Configure basic settings

2 Size
Choose virtual machine size

3 Settings
Configure optional features

4 Summary
NetScaler 11.1 VPX Bring Your ...

5 Buy

Basics

* Name: Citrix-NetScaler-User

VM disk type: SSD

* User name: CitrixUser1

* Authentication type: SSH public key | **Password**

* Password: ••••••••••••••

* Confirm password: ••••••••••••••

Subscription: Microsoft Azure Enterprise

* Resource group: ○ Create new ● Use existing

NetScalerResGroup

Location: Southeast Asia

4. In the **Create virtual machine** pane, specify the required values in each section to create a virtual machine. Click **OK** in each section to save your configuration.

Basic

- Name - specify a name for the NetScaler VPX instance
- VM disk type - select SSD (default value) or HDD from the drop-down menu

- User name and Password - specify a user name and password to access the resources in the resource group that you have created
- Authentication Type - select SSH Public Key or Password
- Resource group - select the resource group you have created from the drop-down list

You can create a resource group here, but Citrix recommends that you create a resource group from Resource groups in Azure Resource Manager and then select the group from the drop-down list.

Size

Depending on the VM disk type, SDD or HDD, you selected in Basic settings, the disk sizes are displayed.

- Select a disk size according to your requirement and click **Select**.

Settings

- Select the default (Standard) disk type
- Storage account - select the storage account
- Virtual network - select the virtual network
- Subnet - set the subnet address
- Public IP address - select the type of IP address assignment
- Network security group - select the security group that you have created. Ensure that inbound and outbound rules are configured in the security group.
- Availability Set - select the availability set from the drop-down box

Summary

The configuration settings are validated and the Summary page displays the result of the validation. If the validation fails, the Summary page displays the reason of the failure. Go back to the particular section and make changes as required. If the validation passes, click **OK**.

Buy

Review the offer details and legal terms on the Purchase page and click **Purchase**.

For high availability deployment, create two independent instances of NetScaler VPX in the same availability set and in the same resource group to deploy them in active-standby and active-active configuration.

Configure multiple IP addresses for a NetScaler VPX standalone instance

September 21, 2020

This section explains how to configure a standalone NetScaler VPX instance with multiple IP addresses, in Azure Resource Manager (ARM). The VPX instance can have one or more NIC attached to it,

and each NIC can have one or more static or dynamic public and private IP addresses assigned to it. You can assign multiple IP addresses as NSIP, VIP, SNIP, and so on.
In a NetScaler instance on Azure, you can assign multiple IP addresses as NSIP, VIP, SNIP, and so on.

For more information, see the Azure documentationAssign multiple IP addresses to virtual machines using the Azure portal.

If you want to use PowerShell commands, see Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands.

Use Case

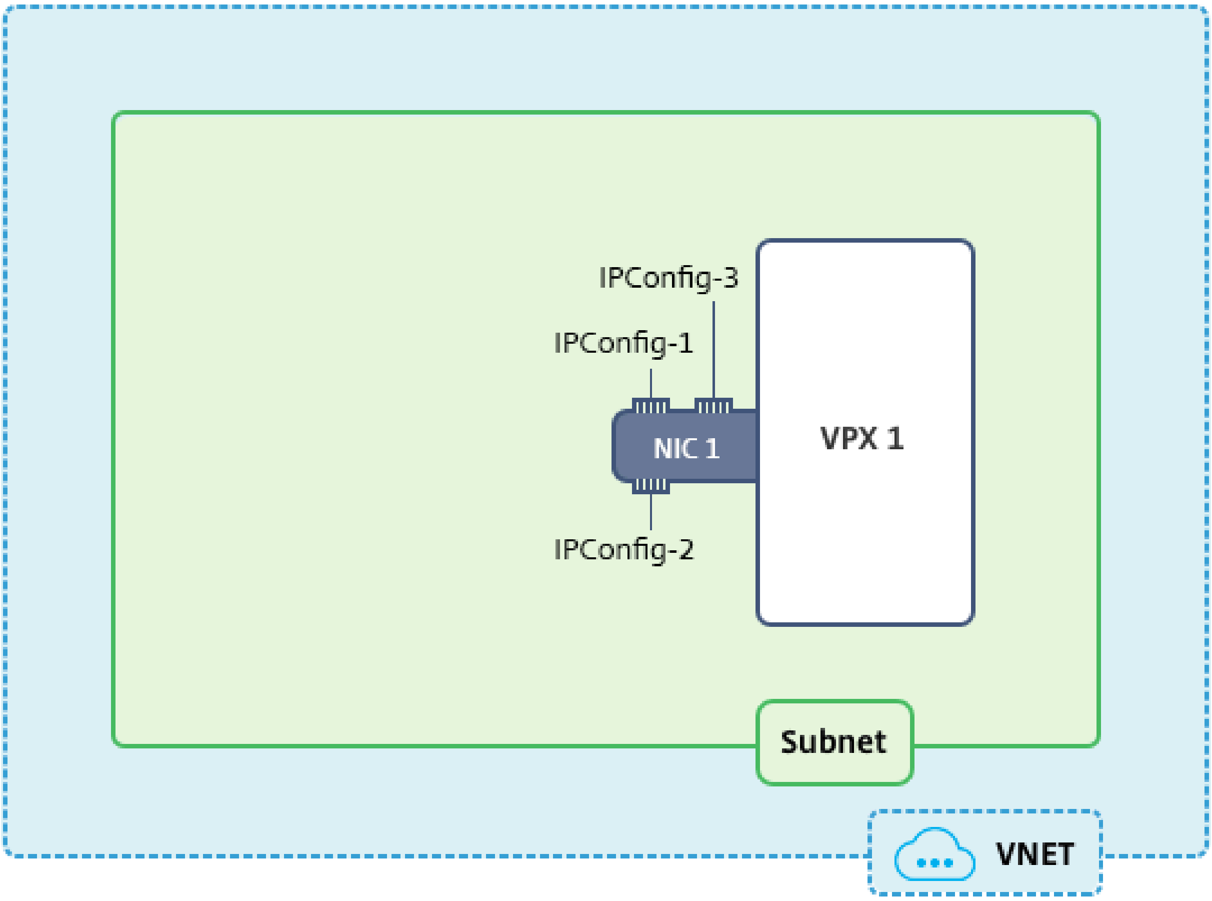
In this use case, a standalone NetScaler VPX appliance is configured with a single NIC that is connected to a virtual network (VNET). The NIC is associated with three IP configurations (ipconfig), each servers a different purpose - as shown in the table.

IPConfig	Assocaited with
IPConfig-1	Static public IP address; static private IP address
IPConfig-2	Static public IP address; static private address
IPConfig-3	Static private IP addres

> **Note**
>
> IPConfig-3 is not associated with any public IP address.

Diagram: Topology

Here is the visual representation of the use case.



> **Note**
>
> In a multi-NIC, multi-IP Azure NetScaler VPX deployment, the private IP associated with the primary (first) IPConfig of the primary (first) NIC is automatically added as the management NSIP of the appliance. The remaining private IP addresses associated with IPConfigs need to be added in the NetScaler appliance as a VIP or SNIP by using the "add ns ip" command, according to your requirement.

Before you begin

Before you begin, create a VPX instance by following the steps given at this link:

Configure a NetScaler VPX standalone instance

For this use case, the NSDoc0330VM NetScaler instance is created.

Procedure to configure multiple IP addresses for a NetScaler VPX instance in standalone mode.

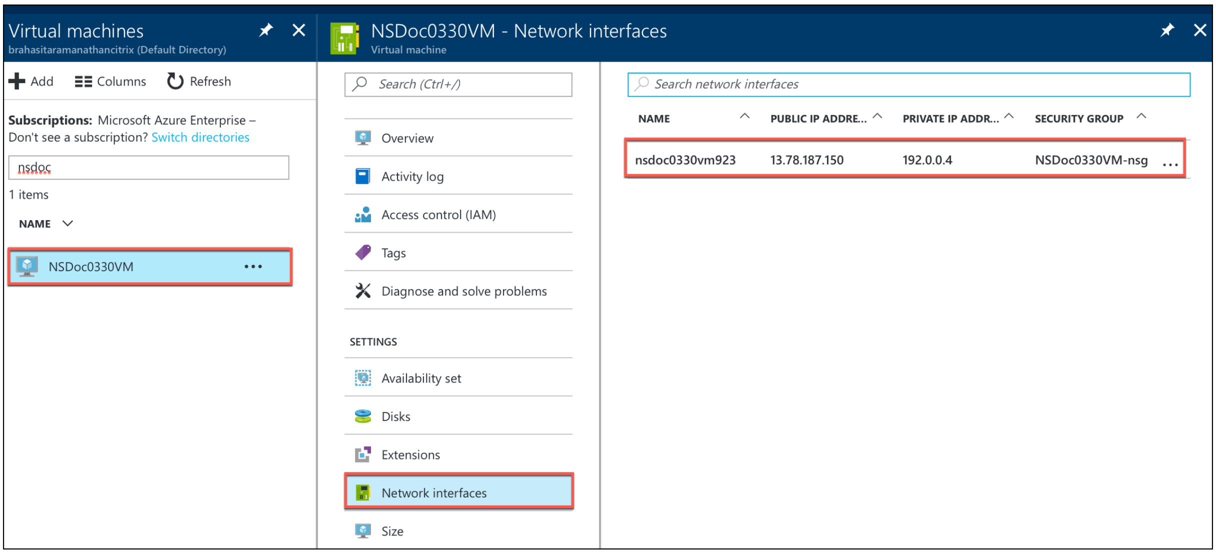
For configuring multiple IP addresses for a NetScaler VPX appliance in standalone mode:

1. Add IP addresses to the VM
2. Configure NetScaler-owned IP addresses

Step 1: Add IP addresses to the VM

1. In the portal, click **More services > type virtual machines** in the filter box, and then click **Virtual machines**.

2. In the **Virtual machines** blade, click the VM you want to add IP addresses to. Click **Network interfaces** in the virtual machine blade that appears, and then select the network interface.


Virtual machines
brahasitaramanathancitrix (Default Directory)
Add
Columns
Refresh
NAME
NSDoc0330VM
NSDoc0330VM - Network interfaces
Virtual machine
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
Diagnose and solve problems
SETTINGS
Availability set
Disks
Extensions
Network interfaces
Size
Search network interfaces

Subscriptions: Microsoft Azure Enterprise – Don't see a subscription? Switch directories

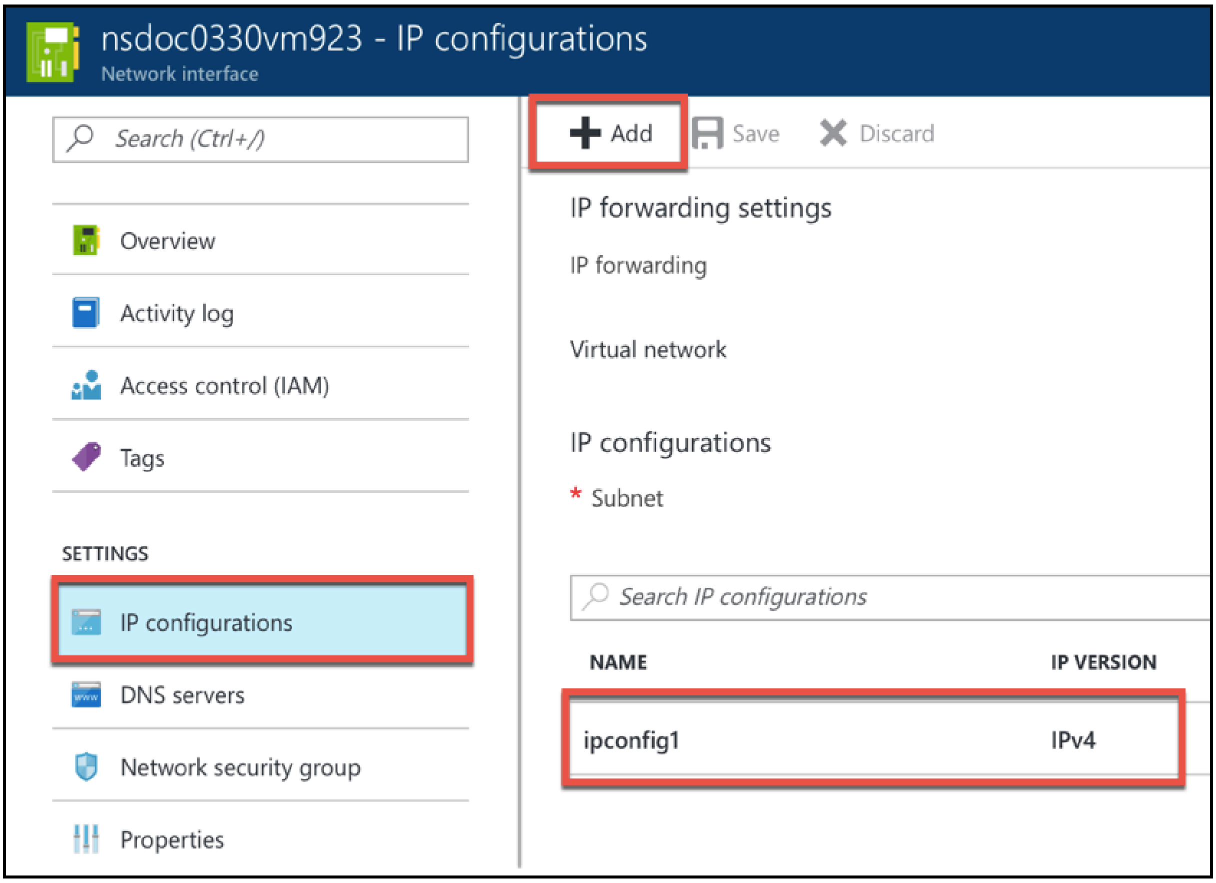
nsdoc

1 items

NAME	PUBLIC IP ADDRE...	PRIVATE IP ADDR...	SECURITY GROUP
nsdoc0330vm923	13.78.187.150	192.0.0.4	NSDoc0330VM-nsg

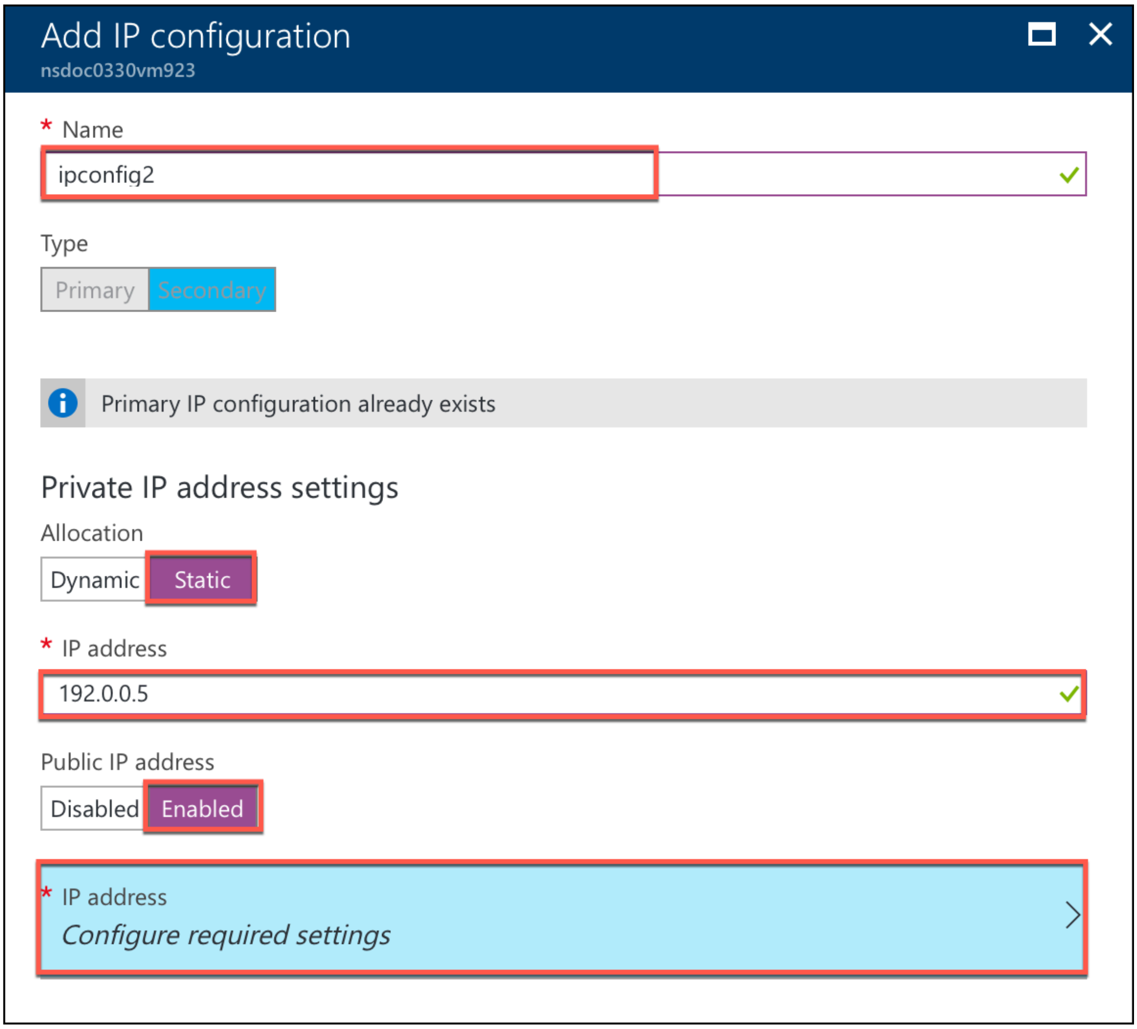
In the blade that appears for the NIC you selected, click **IP configurations**. The existing IP configuration that was assigned when you created the VM, **ipconfig1**, is displayed. For this use case, make sure the IP addresses associated with ipconfig1 are static. Next, create two more IP configurations: ipconfig2 (VIP) and ipconfig3 (SNIP).

To create additional ipconfigs, create **Add**.


nsdoc0330vm923 - IP configurations
Network interface
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
SETTINGS
IP configurations
DNS servers
Network security group
Properties
Add
Save
Discard
IP forwarding settings
IP forwarding
Virtual network
IP configurations
* Subnet
Search IP configurations
NAME
IP VERSION
ipconfig1
IPv4

In the **Add IP configuration** window, enter a **Name**, specify allocation method as **Static**, enter an IP address (192.0.0.5 for this use case), and enable **Public IP address**.

> **Note**
>
> Before adding a static private IP address, check for IP address availability and make sure the IP address belongs to the same subnet to which the NIC is attached.



Add IP configuration
nsdoc0330vm923

* Name
ipconfig2

Type
Primary | Secondary

Primary IP configuration already exists

Private IP address settings

Allocation
Dynamic | Static

* IP address
192.0.0.5

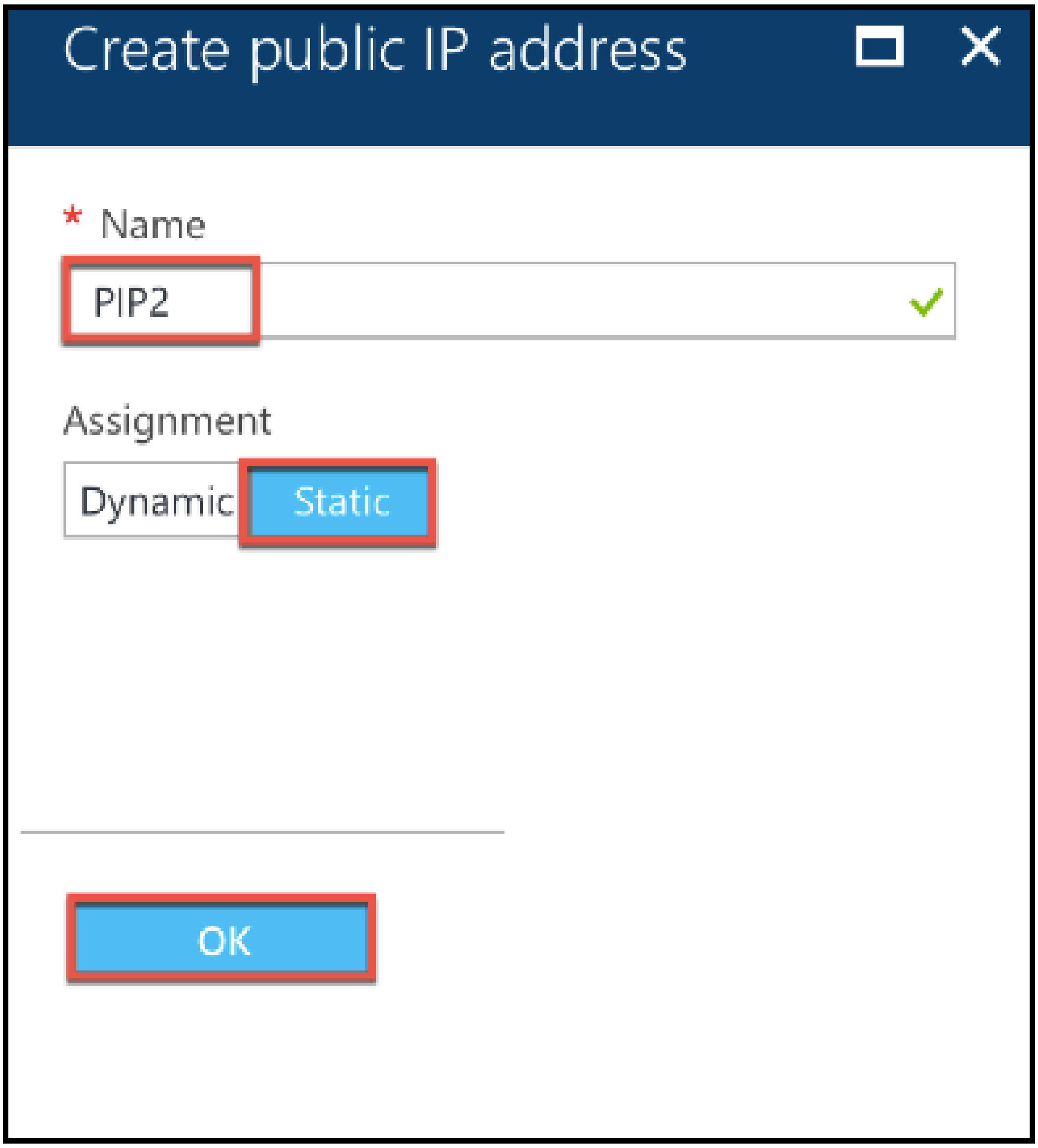
Public IP address
Disabled | Enabled

* IP address
Configure required settings

Next, click **Configure required settings** to create a static public IP addresss for ipconfig2.

By default, public IPs are dynamic. To make sure that the VM always uses the same public IP address, create a static Public IP.

In the Create public IP address blade, add a Name, under Assignment click **Static**. And then click **OK**.



Create public IP address

* Name

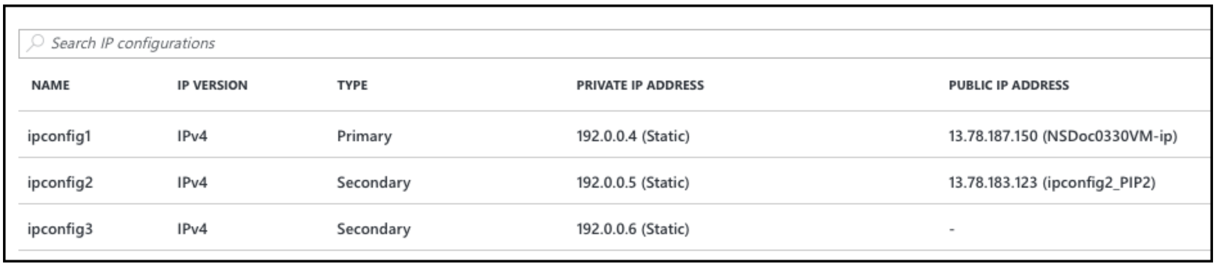
PIP2

Assignment

Dynamic Static

> **Note**
>
> Even when you set the allocation method to static, you cannot specify the actual IP address assigned to the public IP resource. Instead, it gets allocated from a pool of available IP addresses in the Azure location the resource is created in.

Follow the steps to add one more IP configuration for ipconfig3. Public IP is not mandatory.



NAME	IP VERSION	TYPE	PRIVATE IP ADDRESS	PUBLIC IP ADDRESS
ipconfig1	IPv4	Primary	192.0.0.4 (Static)	13.78.187.150 (NSDoc0330VM-ip)
ipconfig2	IPv4	Secondary	192.0.0.5 (Static)	13.78.183.123 (ipconfig2_PIP2)
ipconfig3	IPv4	Secondary	192.0.0.6 (Static)	-

Step 2: Configure NetScaler-owned IP addresses

Configure the NetScaler-owned IP addresses by using the NetScaler GUI or the command "add ns ip." For more information, see https://docs.citrix.com/en-us/netscaler/11-1/networking/ip-addressing/configuring-netscaler-owned-ip-addresses.html

You've now configured multiple IP addresses for a NetScaler VPX instance in standalone mode.

Configure GSLB on NetScaler VPX instances

September 21, 2020

NetScaler appliances configured for global server load balancing (GSLB) provide disaster recovery and continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in case of an outage.

This section describes how to enable GSLB on VPX instances on two sites in a Microsoft Azure environment, by using Windows PowerShell commands.

> **Note**
>
> For more information about GSLB, see Global Server Load Balancing.

You can configure GSLB on a NetScaler VPX instances on Azure, in two steps:

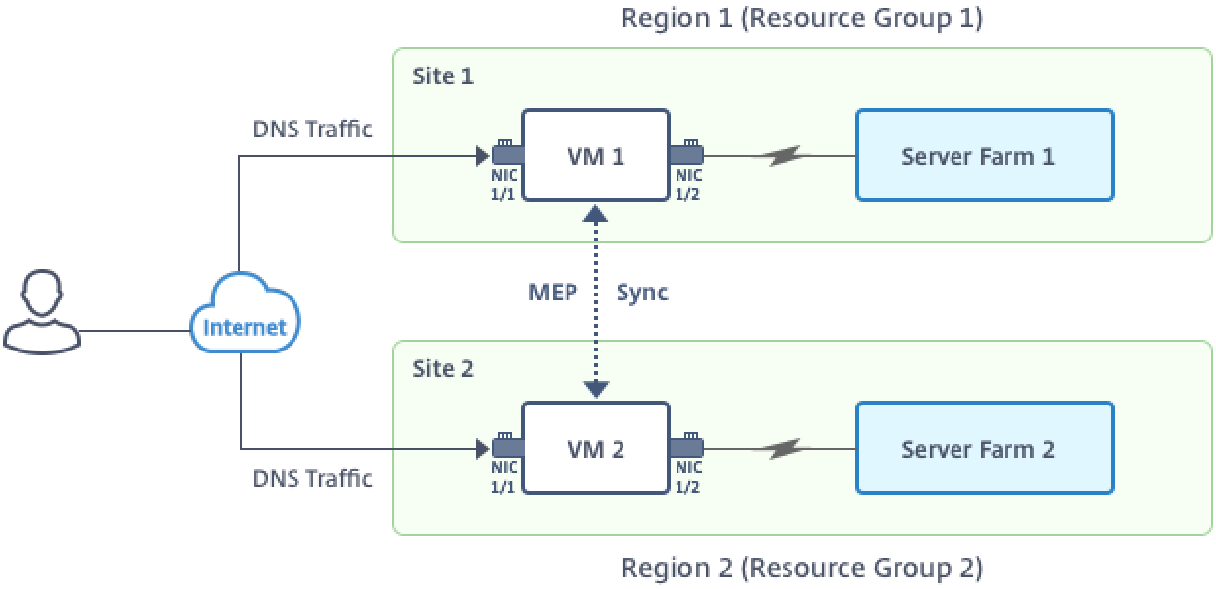
1. Create a VPX instance with multiple NICs and multiple IP addresses, on each site.
2. Enable GSLB on the VPX instances.

> **Note**
>
> For more information about configuring multiple NICs and IP addresses see:
>
> Configuring Multiple IPs for a NetScaler VPX Appliance in Standalone Mode.

Use Case

This use case includes two sites - Site 1 and Site 2. Each site has a VM (VM1 and VM2) configured with multiple NICs, multiple IP addresses, and GSLB.

Figure. GSLB setup implemented across two sites - Site 1 and Site 2.


Region 1 (Resource Group 1)
Site 1
DNS Traffic
NIC 1/1
VM 1
NIC 1/2
Server Farm 1
MEP
Sync
Internet
Site 2
NIC 1/1
VM 2
NIC 1/2
Server Farm 2
DNS Traffic
Region 2 (Resource Group 2)

In this use case, each VM has three NICs - NIC 0/1, 1/1, and 1/2. Each NIC can have multiple private and public IP addresses. The NICs are configured for the following purposes.

- NIC 0/1: to serve management traffic
- NIC 1/1: to serve client-side traffic
- NIC 1/2: to communicate with back-end servers

For information about the IP addresses configured on each NIC in this use case, see the IP Configuration Details section.

Parameters

Following are sample parameters settings for this use case in this document. You can use different settings if you wish.

$location=”West Central US”

$vnetName=”NSVPX-vnet”

$RGName=”multiIP-RG”

$prmStorageAccountName=”multiipstorageaccnt”

$avSetName=”MultiIP-avset”

$vmSize=”Standard_DS3_V2”

Note: The minimum requirement for a VPX instance is 2 vCPUs and 2GB RAM.

$publisher=”citrix”

$offer=”netscalervpx111”

```
$sku=”netscalerbyol”
$version=”latest”
$vmNamePrefix=”MultiIPVPX”
$nicNamePrefix=”MultiipVPX”
$osDiskSuffix=”osdiskdb”
$numberOfVMs=1
$ipAddressPrefix=”10.0.0.”
$ipAddressPrefix1=”10.0.1.”
$ipAddressPrefix2=”10.0.2.”
$pubIPName1=”MultiIP-pip1”
$pubIPName2=”MultiIP-pip2”
$IpConfigName1=”IPConfig1”
$IPConfigName2=”IPConfig-2”
$IPConfigName3=”IPConfig-3”
$IPConfigName4=”IPConfig-4”
$frontendSubnetName=”default”
$backendSubnetName1=”subnet_1”
$backendSubnetName2=”subnet_2”
$suffixNumber=10
```

1. Create a multi-NIC, multi-IP VM by Using PowerShell commands

Follow steps 1-10 to create VM1 with multiple NICs and multi IP addresses, by using PowerShell commands:

1. Create resource group
2. Create storage account
3. Create availability set
4. Create virtual network
5. Create public IP address
6. Create NICs
7. Create VM config object

8. Get credentials and set OS properties for the VM

9. Add NICs

10. Specify OS disk and create VM

After you complete all the steps and commands to create VM1, repeat these steps to create VM2 with parameters specific to it.

Create Resource Group

```
New-AzureRMResourceGroup -Name $RGName -Location $location
```

Create Storage Account

```
$prmStorageAccount=New-AzureRMStorageAccount -Name
    $prmStorageAccountName -ResourceGroupName $RGName -Type Standard_LRS
     -Location $location
```

Create Availability Set

```
$avSet=New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName
    $RGName -Location $location
```

Create Virtual Network

1. Add subnets.

```
$subnet1=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontendSubnetName -AddressPrefix "10.0.0.0/24"
$subnet2=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName1 -AddressPrefix "10.0.1.0/24"
$subnet3=New-AzureRmVirtualNetworkSubnetConfig -Name
    $backendSubnetName2 -AddressPrefix "10.0.2.0/24"
```

2. Add virtual network object.

```
$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName
    $RGName -Location $location -AddressPrefix 10.0.0.0/16 -Subnet
    $subnet1, $subnet2, $subnet3
```

3. Retrieve subnets.

```
$frontendSubnet=$vnet.Subnets|?{
 $_.Name -eq $frontendSubnetName }

$backendSubnet1=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName1 }

$backendSubnet2=$vnet.Subnets|?{
 $_.Name -eq $backendSubnetName2 }
```

Create Public IP Address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName $RGName -Location $location -AllocationMethod Dynamic
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName $RGName -Location $location -AllocationMethod Dynamic
```

Create NICs

NIC 0/1

```
$nic1Name=$nicNamePrefix + $suffixNumber + "-Mgmnt"
$ipAddress1=$ipAddressPrefix + $suffixNumber
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -SubnetId $frontendSubnet.Id -PublicIpAddress $pip1 -PrivateIpAddress $ipAddress1 -Primary
$nic1=New-AzureRMNetworkInterface -Name $nic1Name -ResourceGroupName $RGName -Location $location -IpConfiguration $IpConfig1
```

NIC 1/1

```
$nic2Name $nicNamePrefix + $suffixNumber + "-frontend"
$ipAddress2=$ipAddressPrefix1 + ($suffixNumber)
$ipAddress3=$ipAddressPrefix1 + ($suffixNumber + 1)
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -PublicIpAddress $pip2 -SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress2  -Primary
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -SubnetId $backendSubnet1.Id  -PrivateIpAddress $ipAddress3
nic2=New-AzureRMNetworkInterface -Name $nic2Name -ResourceGroupName $RGName -Location $location -IpConfiguration $IpConfig2, $IpConfig3
```

NIC 1/2

```
$nic3Name=$nicNamePrefix + $suffixNumber + "-backend"
$ipAddress4=$ipAddressPrefix2 + ($suffixNumber)
$IPConfig4=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName4 -SubnetId $backendSubnet2.Id -PrivateIpAddress $ipAddress4 -Primary
$nic3=New-AzureRMNetworkInterface -Name $nic3Name -ResourceGroupName $RGName -Location $location -IpConfiguration $IpConfig4
```

Create VM config object

```
$vmName=$vmNamePrefix
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avSet.Id
```

Get credentials and set OS properties for the VM

```
$cred=Get-Credential -Message "Type the name and password for VPX login."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName $publisher -Offer $offer -Skus $sku -Version $version
```

Add NICs

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -Primary
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic2.Id
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic3.Id
```

Specify OS disk and create VM

```
$osDiskName=$vmName + "-" + $osDiskSuffix
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/" +$osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -Name $sku
```

```
New-AzureRMVM -VM $vmConfig -ResourceGroupName $RGName -Location
    $location
```

> **Note**
>
> Repeat steps 1-10 listed in "Create Multi-NIC VMs by Using PowerShell Commands" to create VM2 with parameters specific to VM2.

IP Configuration details

In this use case, the following IP addresses are used.

Table 1. IP addresses used in VM1

NIC	Private IP	Public IP (PIP)	Description
0/1	10.0.0.10	PIP1	Configured as NSIP (management IP)
1/1	10.0.1.10	PIP2	Configured as SNIP/GSLB Site IP
1/1	10.0.1.11		Configured as LB server IP; public IP is not mandatory
1/2	10.0.2.10		Configured as SNIP for sending monitor probes to services; public IP is not mandatory

Table 2. IP addresses used in VM2

NIC	Private IP	Public IP (PIP)	Description
—	—	—	—
0/1	20.0.0.10	PIP4	Configured as NSIP (management IP)
1/1	20.0.1.10	PIP5	Configured as SNIP/GSLB Site IP
1/1	20.0.1.11		Configured as LB server IP; public IP is not mandatory
1/2	20.0.2.10		Configured as SNIP for sending monitor probes to services; public IP is not mandatory

Here are sample configurations for this use case, showing the IP addresses and intial LB configurations as created through the NetScaler CLI for VM1 and VM2.

```
add ns ip 10.0.1.10 255.255.255.0 -mgmtAccess ENABLED
Add nsip 10.0.2.10 255.255.255.0
```

```
add service svc1 10.0.1.10 ADNS 53
add lb vserver v1 HTTP 10.0.1.11 80
add service s1 10.0.2.120 http 80
Add service s2 10.0.2.121 http 80
Bind lb vs v1 s[1-2]

add ns ip 20.0.1.10 255.255.255.0  -mgmtAccess ENABLED
Add nsip 20.0.2.10 255.255.255.0
add service svc1 20.0.1.10 ADNS 53
add lb vserver v1 HTTP 20.0.1.11 80
Add service s1 20.0.2.90 http 80
Add service s2 20.0.2.91 http 80
Bind lb vs v1 s[1-2]
```

2. Configure GSLB sites and other necessary GSLB settings

Perform the tasks described in the following topic to configure the two GSLB sites and other necessary settings:

Configuring Global Server Load Balancing (GSLB)

For more information, see this support article:
`https://support.citrix.com/article/CTX110348`

Here is a sample GSLB configuration for this use case.

```
enable ns feature LB  GSLB
add gslb site site1 10.0.1.10 -publicIP PIP2
add gslb site site2 20.0.1.10 -publicIP PIP5
add gslb service site1_gslb_http_svc1 10.0.1.11 HTTP 80 -publicIP PIP3
    -publicPort 80 -siteName site1
add gslb service site2_gslb_http_svc1 20.0.1.11 HTTP 80 -publicIP PIP6
    -publicPort 80 -siteName site2
add gslb vserver gslb_http_vip1 HTTP
bind gslb vserver gslb_http_vip1 -serviceName site2_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -serviceName site1_gslb_http_svc1
bind gslb vserver gslb_http_vip1 -domainName www.gslbindia.com -TTL 5
```

You've configured GSLB on NetScaler VPX instances running on Azure. For additional information about how to configure GSLB on NetScaler VPX instances, see the Configuring Citrix NetScaler GSLB in Microsoft Azure video.

Configure address pools (IIP) for a Citrix Gateway appliance

September 21, 2020

In some situations, users who connect with the NetScaler Gateway Plug-in need a unique IP address for NetScaler Gateway. When you enable address pools (also known as IP pooling) for a group, NetScaler Gateway can assign a unique IP address alias to each user. You configure address pools by using intranet IP (IIP) addresses.

You can configure address pools on a NetScaler Gateway deployed on Azure by following this two-step procedure:

- Registering the private IP addresses that will be used in the address pool, in Azure
- Configuring address pools in the NetScaler Gateway appliance

Register a private IP address in the Azure portal

In Azure , you can deploy a NetScaler virtual instance with multiple IP addresses. You can add IP addresses to a VPX instance in two ways:

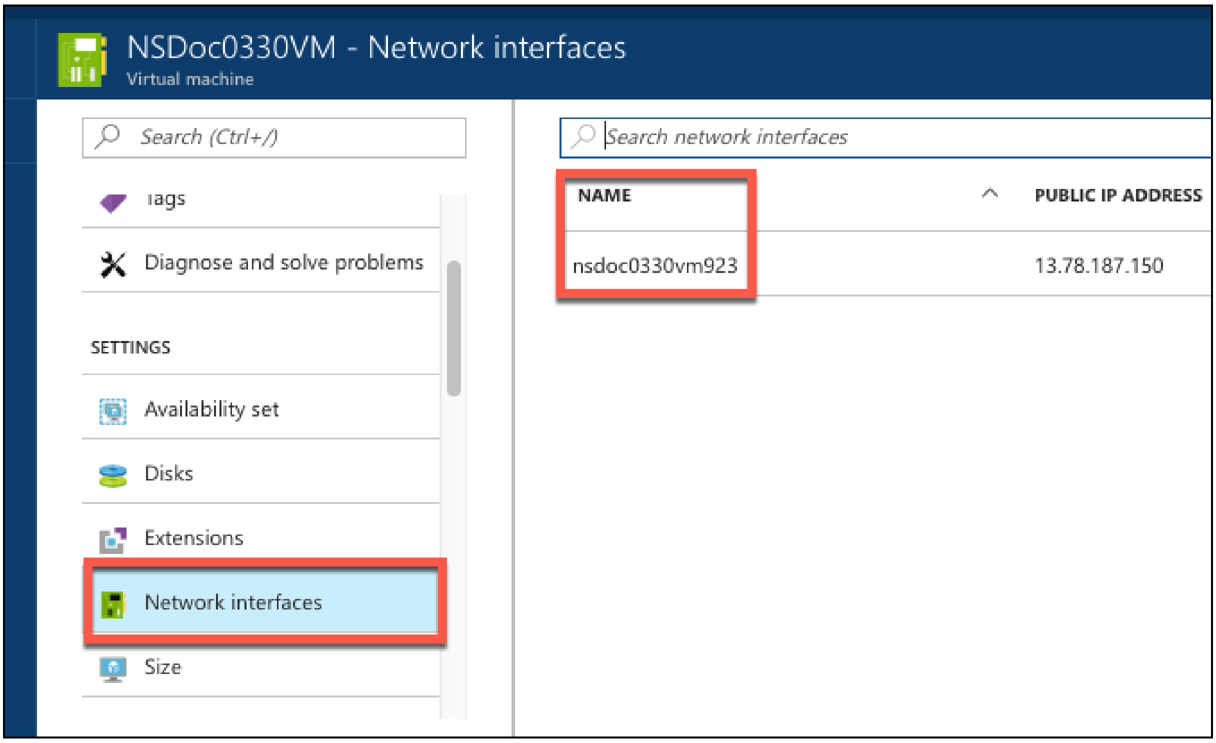
a. While provisioning a VPX instance

For more information about how to add multiple IP addresses while provisioning a VPX instance, see Configuring Multiple IP Addresses for a NetScaler VPX Appliance in Azure Resource Manager. To add IP addresses by using PowerShell commands while provisioning a VPX instance, see Configuring Multiple IP Addresses for a NetScaler VPX Instance in Standalone Mode.

b. After provisioning a VPX instance.

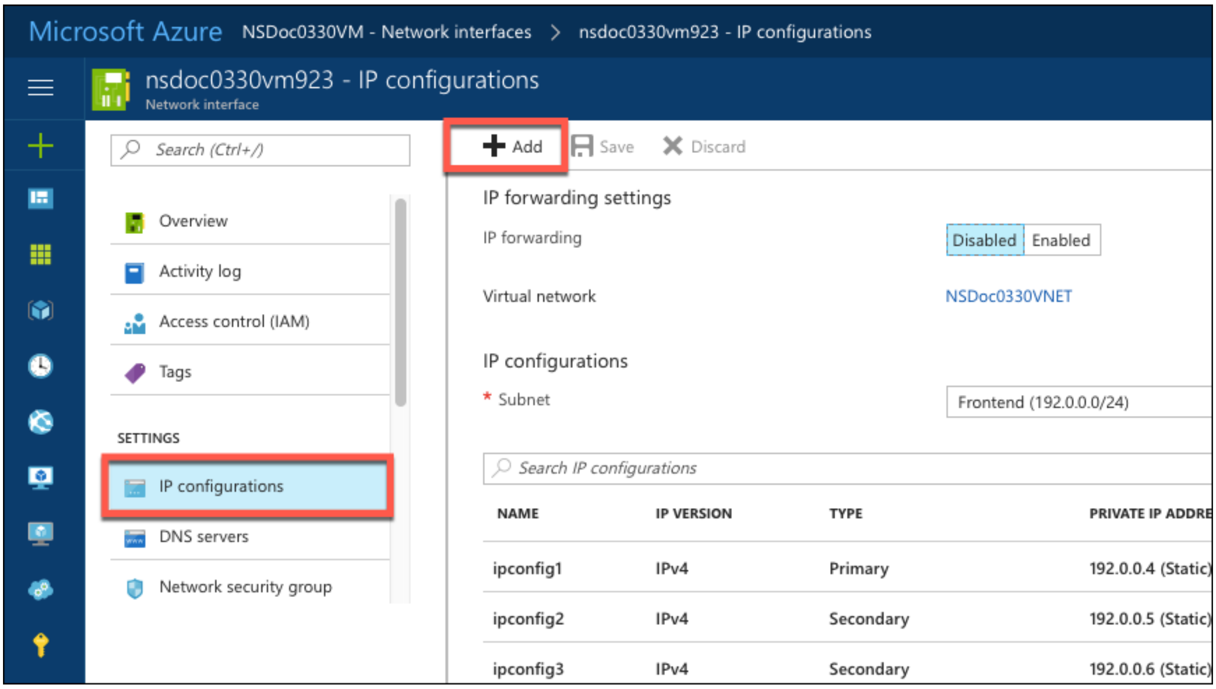
After you've provisioned a VPX instance, follow these steps to register a private IP address in the Azure portal, which you configure as an address pool in the NetScaler Gateway appliance.

1. From Azure Resource Manager (ARM), go to the already created NetScaler VPX instance > **Network interfaces**. Choose the network interface which is bound to a subnet to which the IIP that you want to register belongs.


NSDoc0330VM - Network interfaces
Virtual machine
Search (Ctrl+/)
Tags
Diagnose and solve problems
SETTINGS
Availability set
Disks
Extensions
Network interfaces
Size
Search network interfaces

NAME	PUBLIC IP ADDRESS
nsdoc0330vm923	13.78.187.150

2. Click **IP Configurations**, and then click **Add**.


Microsoft Azure
NSDoc0330VM - Network interfaces > nsdoc0330vm923 - IP configurations
nsdoc0330vm923 - IP configurations
Network interface
Search (Ctrl+/)
Overview
Activity log
Access control (IAM)
Tags
SETTINGS
IP configurations
DNS servers
Network security group
Add
Save
Discard
Search IP configurations

IP forwarding settings

IP forwarding: Disabled | Enabled

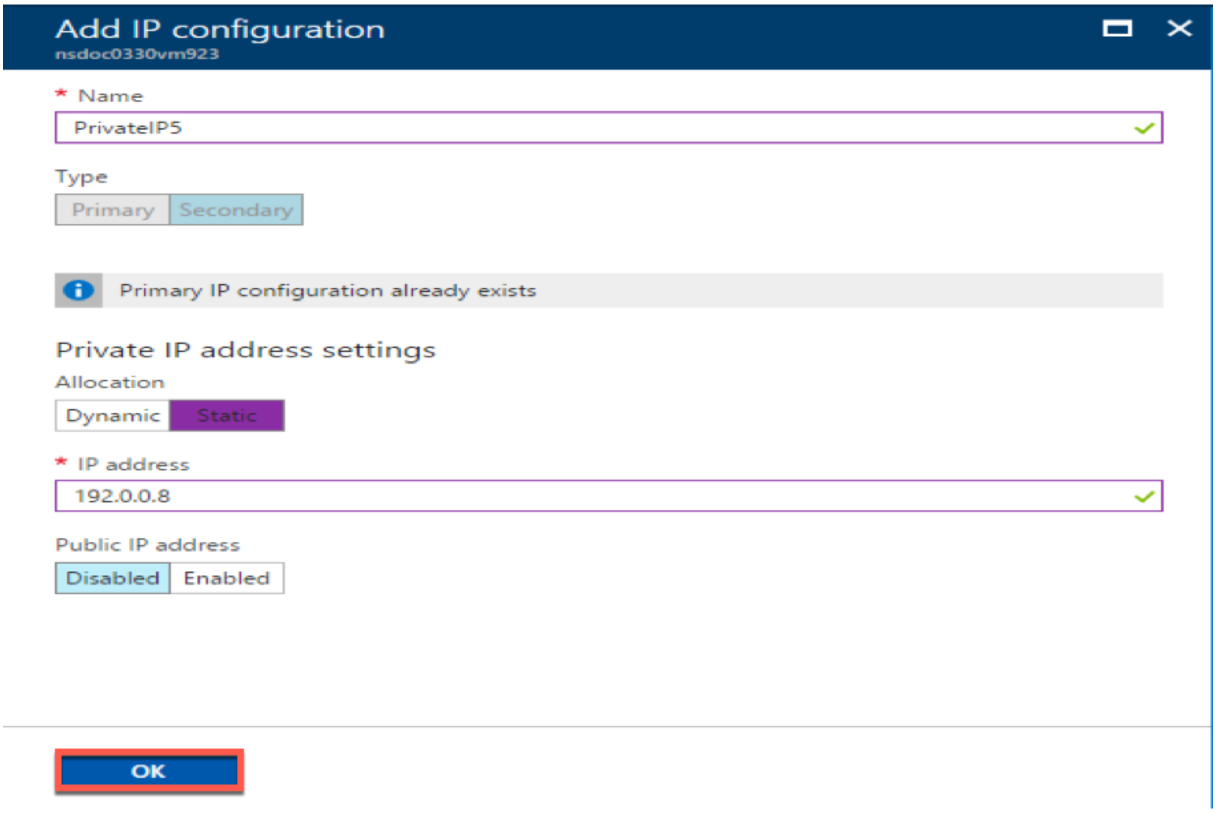
Virtual network: NSDoc0330VNET

IP configurations

* Subnet: Frontend (192.0.0.0/24)

NAME	IP VERSION	TYPE	PRIVATE IP ADDRE
ipconfig1	IPv4	Primary	192.0.0.4 (Static)
ipconfig2	IPv4	Secondary	192.0.0.5 (Static)
ipconfig3	IPv4	Secondary	192.0.0.6 (Static)

3. Provide the required details as shown in the example below and click **OK**.



* Name
PrivateIP5

Type: Secondary

Primary IP configuration already exists

Private IP address settings

Allocation: Static

* IP address
192.0.0.8

Public IP address: Disabled

Configure address pools in the Citrix Gateway appliance

For more information about how to configure address pools on the NetScaler Gateway, see this page.

Limitation: You can not bind a range of IIP addresses to users. Every IIP address that is used in an address pool should be registered.

Configure multiple IP addresses for a NetScaler VPX instance in standalone mode by using PowerShell commands

September 21, 2020

Overview

In an Azure environment, a NetScaler VPX virtual appliance can be deployed with multiple NICs. Each NIC can have multiple IP addresses. This section describes how to deploy a Netscaler VPX instance with a single NIC and multiple IP addresses, by using PowerShell commands. You can use the same script for multi-NIC and multi-IP deployment.

> **Note**
>
> In this document, IP-Config refers to a pair of IP addresses, public IP and private IP, that is associated with an inidivual NIC. For more information, see the Azure Resource Manager Terminology section.

Use case

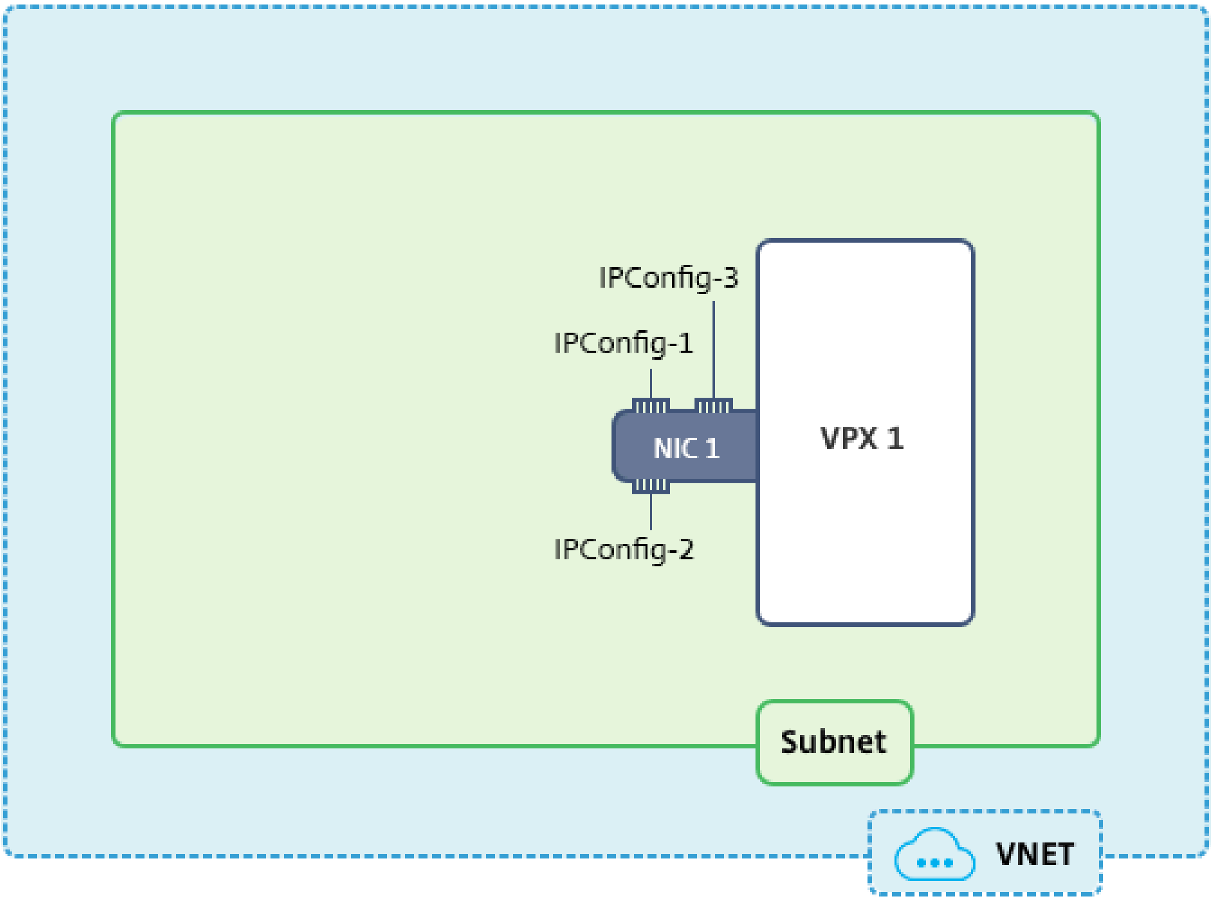
In this use case, a single NIC is connected to a virtual network (VNET). The NIC is associated with three IP configurations, as shown in the following table.

IPConfig	Assocaited with
IPConfig-1	Static public IP address; static private IP address
IPConfig-2	Static public IP address; static private address
IPConfig-3	Static private IP addres

> **Note**
>
> IPConfig-3 is not associated with any public IP address.

Diagram: Topology

Here is the visual representation of the use case.



> **Note**
>
> In a multi-NIC, multi-IP Azure NetScaler VPX deployment, the private IP address associated with the primary (first) IPConfig of the primary (first) NIC is automatically added as the management NSIP address of the appliance. The remaining private IP addresses associated with IPConfigs must be added in the NetScaler appliance as VIPs or SNIPs by using the "add ns ip" command, as determined by your requirements.

Here is the summary of the steps required for configuring multiple IP addresses for a NetScaler VPX virtual appliance in standalone mode:

1. Create Resource Group
2. Create Storage Account
3. Create Availability Set
4. Create NSG
5. Create Virtual Network
6. Create Public IP Address
7. Assign IP Configuration
8. Create NIC
9. Create NetScaler VPX Instance

10. Check NIC Configurations
11. Check VPX-side Configurations

Script

Parameters

Following are sample parameters settings for the use case in this document. You can use different settings if you wish.

$locName=”westcentralus”

$rgName=”Azure-MultiIP”

$nicName1=”VM1-NIC1”

$vNetName=”Azure-MultiIP-vnet”

$vNetAddressRange=”11.6.0.0/16”

$frontEndSubnetName=”frontEndSubnet”

$frontEndSubnetRange=”11.6.1.0/24”

$prmStorageAccountName=”multiipstorage”

$avSetName=”multiip-avSet”

$vmSize=”Standard_DS4_V2” (This parameter creates a VM with upto four NICs.)

Note: The minimum requirement for a VPX instance is 2 vCPUs and 2GB RAM.

$publisher=”citrix”

$offer=”netscalervpx110-6531” (You can use different offers.)

$sku=”netscalerbyol” (According to your offer, the SKU can be different.)

$version=”latest”

$pubIPName1=”PIP1”

$pubIPName2=”PIP2”

$domName1=”multiipvpx1”

$domName2=”multiipvpx2”

$vmNamePrefix=”VPXMultiIP”

$osDiskSuffix=”osmultiipalbdiskdb1”

Network Security Group (NSG)-related information:

$nsgName=”NSG-MultiIP”

$rule1Name=”Inbound-HTTP”

$rule2Name=”Inbound-HTTPS”

$rule3Name=”Inbound-SSH”

$IpConfigName1=”IPConfig1”

$IPConfigName2=”IPConfig-2”

$IPConfigName3=”IPConfig-3”

1. Create Resource Group

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2. Create Storage Account

```
$prmStorageAccount = New-AzureRMStorageAccount -Name $prmStorageAccountName -ResourceGroupName $rgName -Type Standard_LRS -Location $locName
```

3. Create Availability Set

```
$avSet = New-AzureRMAvailabilitySet -Name $avSetName -ResourceGroupName $rgName -Location $locName
```

4. Create Network Security Group (NSG)

1. Add rules. You must add a rule to the NSG for any port that serves traffic.

```
$rule1=New-AzureRmNetworkSecurityRuleConfig -Name $rule1Name -Description "Allow HTTP" -Access Allow -Protocol Tcp -Direction Inbound -Priority 101  -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 80
$rule2=New-AzureRmNetworkSecurityRuleConfig -Name $rule2Name -Description "Allow HTTPS" -Access Allow -Protocol Tcp -Direction Inbound -Priority 110  -SourceAddressPrefix Internet -SourcePortRange * -DestinationAddressPrefix * -DestinationPortRange 443
```

```
$rule3=New-AzureRmNetworkSecurityRuleConfig -Name $rule3Name -
    Description "Allow SSH" -Access Allow -Protocol Tcp -Direction
    Inbound -Priority 120 -SourceAddressPrefix Internet -SourcePortRange
     * -DestinationAddressPrefix * -DestinationPortRange 22
```

2. Create NSG object.

```
$nsg=New-AzureRmNetworkSecurityGroup -ResourceGroupName $rgName -
    Location $locName -Name $nsgName -SecurityRules $rule1,$rule2,$rule3
```

5. Create Virtual Network

1. Add subnets.

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name
    $frontEndSubnetName -AddressPrefix $frontEndSubnetRange
```

2. Add virtual network object.

```
$vnet=New-AzureRmVirtualNetwork -Name $vNetName -ResourceGroupName
    $rgName -Location $locName -AddressPrefix $vNetAddressRange -Subnet
    $frontendSubnet
```

3. Retrieve subnets.

```
$subnetName="frontEndSubnet"
$subnet1=$vnet.Subnets|?{
 $_.Name -eq $subnetName }
```

6. Create Public IP Address

```
$pip1=New-AzureRmPublicIpAddress -Name $pubIPName1 -ResourceGroupName
    $rgName -DomainNameLabel $domName1 -Location $locName -
    AllocationMethod Static
$pip2=New-AzureRmPublicIpAddress -Name $pubIPName2 -ResourceGroupName
    $rgName -DomainNameLabel $domName2 -Location $locName -
    AllocationMethod Static
```

> **Note**
>
> Check availability of domain names before using.
>
> Allocation method for IP addresses can be dynamic or static.

7. Assign IP Configuration

In this use case, consider the following points before assigning IP addresses:

- IPConfig-1 belongs to subnet1 of VPX1.
- IPConfig-2 belongs to subnet 1 of VPX1.
- IPConfig-3 belongs to subnet 1 of VPX1.

> **Note**
>
> When you assign multiple IP configurations to a NIC, one configuration must be assigned as primary.

```
$IPAddress1="11.6.1.27"
$IPConfig1=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName1 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress1 -PublicIpAddress $pip1
      - Primary
$IPAddress2="11.6.1.28"
$IPConfig2=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName2 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress2 -PublicIpAddress $pip2
$IPAddress3="11.6.1.29"
$IPConfig3=New-AzureRmNetworkInterfaceIpConfig -Name $IPConfigName3 -
    Subnet $subnet1 -PrivateIpAddress $IPAddress3 -Primary Use a valid
    IP address that meets your subnet requirements and check its
    availability.
```

8. Create NIC

```
$nic1=New-AzureRmNetworkInterface -Name $nicName1 -ResourceGroupName
    $rgName -Location $locName -IpConfiguration $IpConfig1,$IpConfig2,
    $IPConfig3 -NetworkSecurityGroupId $nsg.Id
```

9. Create NetScaler VPX Instance

1. Initialize variables.

```
$suffixNumber = 1
$vmName = $vmNamePrefix + $suffixNumber
```

2. Create VM config object.

```
$vmConfig=New-AzureRMVMConfig -VMName $vmName -VMSize $vmSize -
    AvailabilitySetId $avSet.Id
```

3. Set credentials, OS, and image.

```
$cred=Get-Credential -Message "Type the name and password for VPX login
    ."
$vmConfig=Set-AzureRMVMOperatingSystem -VM $vmConfig -Linux -
    ComputerName $vmName -Credential $cred
$vmConfig=Set-AzureRMVMSourceImage -VM $vmConfig -PublisherName
    $publisher -Offer $offer -Skus $sku -Version $version
```

4. Add NIC.

```
$vmConfig=Add-AzureRMVMNetworkInterface -VM $vmConfig -Id $nic1.Id -
    Primary
```

> **Note**
>
> In a multi-NIC VPX deployment, one NIC should be primary. So, "-Primary" needs to be appended while adding that NIC to the VPX instance.

5. Specify OS disk and create VM.

```
$osDiskName=$vmName + "-" + $osDiskSuffix1
$osVhdUri=$prmStorageAccount.PrimaryEndpoints.Blob.ToString() + "vhds/"
     + $osDiskName + ".vhd"
$vmConfig=Set-AzureRMVMOSDisk -VM $vmConfig -Name $osDiskName -VhdUri
    $osVhdUri -CreateOption fromImage
Set-AzureRmVMPlan -VM $vmConfig -Publisher $publisher -Product $offer -
    Name $sku
New-AzureRMVM -VM $vmConfig -ResourceGroupName $rgName -Location
    $locName
```

10. Check NIC Configurations

After the VPX instance starts, you can check the IP addresses allocated to IPConfigs of the VPX NIC by using the following command.

```
$nic.IPConfig
```

11. Check VPX-side Configurations

When the Netscaler VPX instance starts, a private IP address associated with primary IPconfig of the primary NIC is added as the NSIP address. The remaining private IP addresses must be added as VIP or SNIP addresses, as determined by your requirements. Use the following command.

```
add nsip <Private IPAddress><netmask> -type VIP/SNIP
```

You've now configured multiple IP addresses for a NetScaler VPX instance in standalone mode. For additional information about how to configure multiple IP addresses for a NetScaler VPX instance in standalone mode, see the Citrix NetScaler deployments in Microsoft Azure video.

PowerShell scripts for Azure deployment

September 21, 2020

This topic provides the PowerShell cmdlets with which you can perform the following configurations in Azure PowerShell:

- Provision NetScaler VPX in Standalone Mode
- Configure NetScaler VPX HA with Azure External Load Balancer
- Configure NetScaler VPX HA with Azure Internal Load Balance

Provision NetScaler VPX in Standalone Mode

1. **Creating a resource group**

The resource group can include all the resources for the solution, or only those resources that you want to manage as a group. The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

$rgName="<resource group name>"

$locName="<location name, such as West US>"

New-AzureRmResourceGroup -Name $rgName -Location $locName

For example:

$rgName = "ARM-VPX"

$locName = "West US"

New-AzureRmResourceGroup -Name $rgName -Location $locName

2. **Creating a storage account**

Choose a unique name for your storage account that contains only lowercase letters and numbers.

$saName="<storage account name>"

$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"

```
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

For example:

```
$saName=”vpxstorage”

$saType=”Standard_LRS”

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

3. **Creating an availability set**

Availability set helps to keep your virtual machines available during downtime, such as during maintenance. A load balancer configured with an availability set ensures that your application is always available.

```
$avName=”<availability set name>”

New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
```

4. **Creating a virtual network**

Add a new virtual network with at least one subnet, if the subnet was not created previously.

```
$FrontendAddressPrefix=”10.0.1.0/24”

$BackendAddressPrefix=”10.0.2.0/24”

$vnetAddressPrefix=”10.0.0.0/16”

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix $BackendAddressPrefix

New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet,$backendSubnet
```

For example:

```
$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix $BackendAddressPrefix

New-AzureRmVirtualNetwork -Name TestNet -ResourceGroupName $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet,$backendSubnet
```

5. **Creating a NIC**

Create a NIC and associate the NIC with the NetScaler VPX instance. The front end Subnet created in the above procedure is indexed at 0 and the back end Subnet is indexed at 1. Now create NIC in one of the three following ways:

a) *NIC with Public IP address*

```
$nicName=”<name of the NIC of the VM>”
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -Location $locName -AllocationMethod Dynamic
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id
```

b) *NIC with Public IP and DNS label*

```
$nicName=”<name of the NIC of the VM>”
```

```
$domName=”<domain name label>”
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod Dynamic
```

Before assigning $domName, check it is available or not by using command:

```
Test-AzureRmDnsAvailability -DomainQualifiedName $domName -Location $locName
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id
```

For example:

```
$nicName=”frontendNIC”
```

```
$domName=”vpxazure”
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -DomainNameLabel $domName -Location $locName -AllocationMethod Dynamic
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[0].Id -PublicIpAddressId $pip.Id
```

c) *NIC with Dynamic Public Address and Static Private IP address*

Make sure that the private (static) IP address you add to the VM should be the same range as that of the subnet specified.

```
$nicName=”<name of the NIC of the VM>”
```

```
$staticIP=”<available static IP address on the subnet>”
```

```
$pip = New-AzureRmPublicIpAddress -Name $nicName -ResourceGroupName $rgName -Location $locName -AllocationMethod Dynamic
```

```
$nic = New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -SubnetId $vnet.Subnets[$subnetIndex].Id -PublicIpAddressId $pip.Id -PrivateIpAddress $staticIP
```

6. **Creating a virtual object**

```
$vmName=”<VM name>”
```

```
$vmSize=”<VM size string>”
```

```
$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName
```

```
$vm=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id
```

7. **Getting the NetScaler VPX image**

```
$pubName=”<Image publisher name>”
```

```
$offerName=”<Image offer name>”
```

```
$skuName=”<Image SKU name>”
```

```
$cred=Get-Credential -Message “Type the name and password of the local administrator account.”
```

Provide your credentials that is used to login into VPX

```
$vm=Set-AzureRmVMOperatingSystem -VM $vm -Linux -ComputerName $vmName -Credential $cred -Verbose
```

```
$vm=Set-AzureRmVMSourceImage -VM $vm -PublisherName $pubName -Offer $offerName -Skus $skuName -Version “latest”
```

```
$vm=Add-AzureRmVMNetworkInterface -VM $vm -Id $nic.Id
```

For example:

```
$pubName=”citrix”
```

The following command is used for displaying all offers from Citrix:

Get-AzureRMVMImageOffer -Location $locName -Publisher $pubName	Select Offer

```
$offerName=”netscalervpx110-6531”
```

The following command is used to know sku offered by publisher for specific offer name:

Get-AzureRMVMImageSku -Location $locName -Publisher $pubName -Offer $offerName	Select Skus

8. **Creating a virtual machine**

```
$diskName=”<name identifier for the disk in Azure storage, such as OSDisk>”
```

For example:

```
$diskName=”dynamic”

$pubName=”citrix”

$offerName=”netscalervpx110-6531”

$skuName=”netscalerbyol”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri=$storageAcc.PrimaryEndpoints.Blob.ToString() + “vhds/” + $diskName  + “.vhd”

$vm=Set-AzureRmVMOSDisk -VM $vm -Name $diskName -VhdUri $osDiskUri -CreateOption fromImage
```

When you create VM from Images present in marketplace, use the following command to specify the VM plan:

```
Set-AzureRmVMPlan -VM $vm -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm
```

Provision NetScaler VPX in HA with Azure External Load Balancer

Log on to AzureRmAccount using your Azure user credentials.

1) **Creating a resource group**

The location specified here is the default location for resources in that resource group. Make sure that all commands used to create a load balancer use the same resource group.

```
$rgName=”<resource group name>”

$locName=”<location name, such as West US>”

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

For example:

```
$rgName = “ARM-LB-NS”

$locName = “West US”

New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2) **Creating a storage account**

Choose a unique name for your storage account that contains only lowercase letters and numbers.

```
$saName=”<storage account name>”

$saType=”<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>”

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

For example:

```
$saName=”vpxstorage”

$saType=”Standard_LRS”

New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

3) **Creating an availability set**

A load balancer configured with an availability set ensures that your application is always available.

```
$avName=”<availability set name>”

New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
```

4) **Creating a virtual network**

Add a new virtual network with at least one subnet, if the subnet was not created previously.

```
$vnetName = “LBVnet”

$FrontendAddressPrefix=”10.0.1.0/24”

$BackendAddressPrefix=”10.0.2.0/24”

$vnetAddressPrefix=”10.0.0.0/16”

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix $BackendAddressPrefix

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet,$backendSubnet‘
```

Note: Choose the AddressPrefix parameter value as per your requirement.

Assign front end and back end subnet to the virtual network that you created earlier in this step.

If the front end subnet is the first element of array vnet, subnetId should be $vnet.Subnets[0].Id.

If the front end subnet is the second element in the array, the subnetId should be $vnet.Subnets[1].Id, and so on..

5) Configuring front end IP address and creating back end address pool

Configure a front end IP address for the incoming load balancer network traffic and create a back end address pool to receive the load balanced traffic.

```
$pubName=”PublicIp1”
```

```
$publicIP1 = New-AzureRmPublicIpAddress -Name $pubName -ResourceGroupName $rgName -Location $locName -AllocationMethod Static -DomainNameLabel nsvpx
```

Note: Check for the availability of the value for DomainNameLabel.

```
$FIPName = “ELBFIP”
```

```
$frontendIP1 = New-AzureRmLoadBalancerFrontendIpConfig -Name $FIPName -PublicIpAddress $publicIP1
```

```
$BEPool = “LB-backend-Pool”
```

```
$beaddresspool1= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name $BEPool
```

8) Creating a health probe

Create a TCP health probe with port 9000 and interval 5 seconds.

```
$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name HealthProbe -Protocol Tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2
```

9) Creating a load balancing rule

Create a LB rule for each service that you are load balancing.

For example:

You can use the following example to load balance http service.

```
$lbrule1 = New-AzureRmLoadBalancerRuleConfig -Name “HTTP-LB” -FrontendIpConfiguration $frontendIP1 -BackendAddressPool  $beAddressPool1 -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80
```

10) Creating inbound NAT rules

Create NAT rules for services that you are not load balancing.

For example, when creating a SSH access to a NetScaler VPX instance.

Note: Protocol-FrontEndPort-BackendPort triplet should not be the same for two NAT rules.

```
$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH1 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 22 -BackendPort 22
```

```
$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name SSH2 -FrontendIpConfiguration $frontendIP1 -Protocol TCP -FrontendPort 10022 -BackendPort 22
```

11) **Creating a load balancer entity**

Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

$lbName=”ELB”

$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgName -Name $lbName -Location $locName -InboundNatRule $inboundNATRule1, $inboundNATRule2 -FrontendIpConfiguration $frontendIP1 -LoadBalancingRule $lbrule1 -BackendAddressPool $beAddressPool1 -Probe $healthProbe

12) **Creating a NIC**

Create two NICs and associate each NIC with each VPX instance

a) NIC1 with VPX1

For example:

$nicName=”NIC1”

$lbName=”ELB”

$bePoolIndex=0

* Rule indexes starts from 0.

$natRuleIndex=0

$subnetIndex=0

* Frontend subnet index

$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic1=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location
$locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$
-LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]

b) NIC2 with VPX2

*For example***:**

$nicName=”NIC2”

$lbName=”ELB”

$bePoolIndex=0

$natRuleIndex=1

* Second Inbound NAT (SSH) rule we need to use

$subnetIndex=0

* Frontend subnet index

```
$lb=Get-AzureRmLoadBalancer -Name $lbName -ResourceGroupName $rgName

$nic2=New-AzureRmNetworkInterface -Name $nicName -ResourceGroupName $rgName -Location $locName -Subnet $vnet.Subnets[$subnetIndex] -LoadBalancerBackendAddressPool $lb.BackendAddressPools[$
-LoadBalancerInboundNatRule $lb.InboundNatRules[$natRuleIndex]
```

13) **Creating NetScaler VPX instances**

Create two NetScaler VPX instances as part of the same resource group and availability set, and attach it to the external load balancer.

a) NetScaler VPX instance 1

For example:

```
$vmName=”VPX1”

$vmSize=”Standard_A3”

$pubName=”citrix”

$offerName=”netscalervpx110-6531”

$skuName=”netscalerbyol”

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message “Type Credentials which will be used to login to VPX instance”

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version “latest”

$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $nic1.Id

$diskName=”dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + “vhds1/” + $diskName + “.vhd”

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

b) NetScaler VPX instance 2

For example:

```
$vmName=”VPX2”
$vmSize=”Standard_A3”
$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName
$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id
$cred=Get-Credential -Message “ Type Credentials which will be used to login to VPX instance “
$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose
$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version “latest”
$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $nic2.Id
$diskName=”dynamic”
$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName
$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + “vhds2/” + $diskName + “.vhd”
$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage
Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName
New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2
```

14) **Configuring the virtual machines**

When both the NetScaler VPX instances start, then connect to both NetScaler VPX instances using the SSH protocol to configure the virtual machines.

a) Active-Active: Run the same set of configuration commands on the command line of both the NetScaler VPX instances.

b) Active-Passive: Run this command on the command line of both the NetScaler VPX instances.

```
add ha node #nodeID <nsip of other NetScaler VPX>
```

In Active-Passive mode, run configuration commands on the primary node only.

Provision NetScaler VPX in HA with Azure Internal Load Balancer

Log on to AzureRmAccount using your Azure user credentials.

1) **Creating a resource group**

The location specified here is the default location for resources in that resource group. Make sure all commands to create a load balancer use the same resource group.

```
$rgName="<resource group name>"
```

```
$locName="<location name, such as West US>"
```

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

For example:

```
$rgName = "ARM-LB-NS"
```

```
$locName = "West US"
```

```
New-AzureRmResourceGroup -Name $rgName -Location $locName
```

2) **Creating a storage account**

Choose a unique name for your storage account that contains only lowercase letters and numbers.

```
$saName="<storage account name>"
```

```
$saType="<storage account type, specify one: Standard_LRS, Standard_GRS, Standard_RAGRS, or Premium_LRS>"
```

```
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

For example:

```
$saName="vpxstorage"
```

```
$saType="Standard_LRS"
```

```
New-AzureRmStorageAccount -Name $saName -ResourceGroupName $rgName -Type $saType -Location $locName
```

3) **Creating an availability set**

A load balancer configured with an availability set ensures that your application is always available..

```
$avName="<availability set name>"
```

```
New-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName -Location $locName
```

4) **Creating a virtual network**

Add a new virtual network with at least one subnet, if the subnet was not created previously.

```
$vnetName = "LBVnet"
```

```
$vnetAddressPrefix="10.0.0.0/16"
```

```
$FrontendAddressPrefix="10.0.1.0/24"
```

$BackendAddressPrefix=”10.0.2.0/24”

$vnet=New-AzureRmVirtualNetwork -Name $vnetName -ResourceGroupName $rgName -Location $locName -AddressPrefix $vnetAddressPrefix -Subnet $frontendSubnet,$backendSubnet‘

$frontendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix $FrontendAddressPrefix

$backendSubnet=New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix $BackendAddressPrefix

Note: Choose the AddressPrefix parameter value as per your requirement.

Assign front end and back end subnet to the virtual network that you created earlier in this step.

If the front end subnet is the first element of array vnet, subnetId should be $vnet.Subnets[0].Id.

If the front end subnet is the second element in the array, the subnetId should be $vnet.Subnets[1].Id, and so on..

5) **Creating an back end address pool**

$beaddresspool= New-AzureRmLoadBalancerBackendAddressPoolConfig -Name “LB-backend”

6) **Creating NAT rules**

Create NAT rules for services that you are not load balancing.

$inboundNATRule1= New-AzureRmLoadBalancerInboundNatRuleConfig -Name “Inboundnatrule1” -FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3441 -BackendPort 3389

$inboundNATRule2= New-AzureRmLoadBalancerInboundNatRuleConfig -Name “RDP2” -FrontendIpConfiguration $frontendIP -Protocol TCP -FrontendPort 3442 -BackendPort 3389

Use front end and back end ports as per your requirement.

7) **Creating a health probe**

Create a TCP health probe with port 9000 and interval 5 seconds.

$healthProbe = New-AzureRmLoadBalancerProbeConfig -Name “HealthProbe” ” -Protocol tcp -Port 9000 -IntervalInSeconds 5 -ProbeCount 2

8) **Creating a load balancing rule**

Create a LB rule for each service that you are load balancing.

For example:

You can use the following example to load balance http service.

$lbrule = New-AzureRmLoadBalancerRuleConfig -Name “lbrule1” -FrontendIpConfiguration $frontendIP -BackendAddressPool $beAddressPool -Probe $healthProbe -Protocol Tcp -FrontendPort 80 -BackendPort 80

Use front end and back end ports as per your requirement.

9) **Creating a load balancer entity**

Create the load balancer adding all objects (NAT rules, load balancer rules, probe configurations) together.

```
$NRPLB = New-AzureRmLoadBalancer -ResourceGroupName $rgname -Name “InternalLB” -Location $locName -FrontendIpConfiguration $frontendIP -InboundNatRule $inboundNATRule1,$inboundNatRule2 -LoadBalancingRule $lbrule -BackendAddressPool $beAddressPool -Probe $healthProbe
```

10) **Creating a NIC**

Create two NICs and associate each NIC with each NetScaler VPX instance

```
$backendnic1= New-AzureRmNetworkInterface -ResourceGroupName $rgName -Name lb-nic1-be -Location $locName -PrivateIpAddress 10.0.2.6 -Subnet $backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule $nrplb.InboundNatRules[0]
```

This NIC is for NetScaler VPX 1. The Private IP should be in same subnet as that of subnet added.

```
$backendnic2= New-AzureRmNetworkInterface -ResourceGroupName $rgName -Name lb-nic2-be -Location $locName -PrivateIpAddress 10.0.2.7 -Subnet $backendSubnet -LoadBalancerBackendAddressPool $nrplb.BackendAddressPools[0] -LoadBalancerInboundNatRule $nrplb.InboundNatRules[1].
```

This NIC is for NetScaler VPX 2.The parameter Private IPAddress can have any private IP as per your requirement.

11) **Creating NetScaler VPX instances**

Create two VPX instances part of same resource group and availability set and attach it to the internal load balancer.

a) NetScaler VPX instance 1

For example:

```
$vmName=”VPX1”

$vmSize=”Standard_A3”

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm1=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message “Type Credentials which will be used to login to VPX instance”

$vm1=Set-AzureRmVMOperatingSystem -VM $vm1 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm1=Set-AzureRmVMSourceImage -VM $vm1 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version “latest”
```

```
$vm1=Add-AzureRmVMNetworkInterface -VM $vm1 -Id $backendnic1.Id

$diskName=”dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + “vhds1/” + $diskName  + “.vhd”

$vm1=Set-AzureRmVMOSDisk -VM $vm1 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm1 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm1
```

b) NetScaler VPX instance 2

For example:

```
$vmName=”VPX2”

$vmSize=”Standard_A3”

$avSet=Get-AzureRmAvailabilitySet -Name $avName -ResourceGroupName $rgName

$vm2=New-AzureRmVMConfig -VMName $vmName -VMSize $vmSize -AvailabilitySetId $avset.Id

$cred=Get-Credential -Message “ Type Credentials which will be used to login to VPX instance “

$vm2=Set-AzureRmVMOperatingSystem -VM $vm2 -Linux -ComputerName $vmName -Credential $cred -Verbose

$vm2=Set-AzureRmVMSourceImage -VM $vm2 -PublisherName $pubName -Offer $offerName -Skus $skuName -Version “latest”

$vm2=Add-AzureRmVMNetworkInterface -VM $vm2 -Id $backendnic2.Id

$diskName=”dynamic”

$storageAcc=Get-AzureRmStorageAccount -ResourceGroupName $rgName -Name $saName

$osDiskUri1=$storageAcc.PrimaryEndpoints.Blob.ToString() + “vhds2/” + $diskName  + “.vhd”

$vm2=Set-AzureRmVMOSDisk -VM $vm2 -Name $diskName -VhdUri $osDiskUri1 -CreateOption fromImage

Set-AzureRmVMPlan -VM $vm2 -Publisher $pubName -Product $offerName -Name $skuName

New-AzureRmVM -ResourceGroupName $rgName -Location $locName -VM $vm2
```

12) **Configuring the virtual machines**

When both the NetScaler VPX instances start, then connect to both NetScaler VPX instances using the SSH protocol to configure the virtual machines.

a) Active-Active: Run the same set of configuration commands on the command line of both the NetScaler VPX instances.

b) Active-Passive: Run this command on the command line of both the NetScaler VPX instances.

```
add ha node #nodeID <nsip of other NetScaler VPX>
```

In Active-Passive mode, run configuration commands on the primary node only.

Azure terminology

September 21, 2020

Some of the new terms used in Azure Resource Manager are listed below. This list helps you relate to the terms and concepts used in Azure Service Management or classical mode of deployment.

1. Azure Load Balancer – Azure load balancer is a resource that distributes incoming traffic among computers in a network. Traffic is distributed among virtual machines defined in a load-balancer set. A load balancer can be external or internet-facing, or it can be internal.

2. Azure Resource Manager (ARM) – ARM is the new management framework for services in Azure. Azure Load Balancer is managed using ARM-based APIs and tools.

3. Back-End Address Pool – These are IP addresses associated with the virtual machine Network Interface Card (NIC) to which load will be distributed.

4. BLOB - Binary Large Object – Any binary object like a file or an image that can be stored in Azure storage.

5. Front-End IP Configuration – An Azure Load balancer can include one or more front-end IP addresses, also known as a virtual IPs (VIPs). These IP addresses serve as ingress for the traffic.

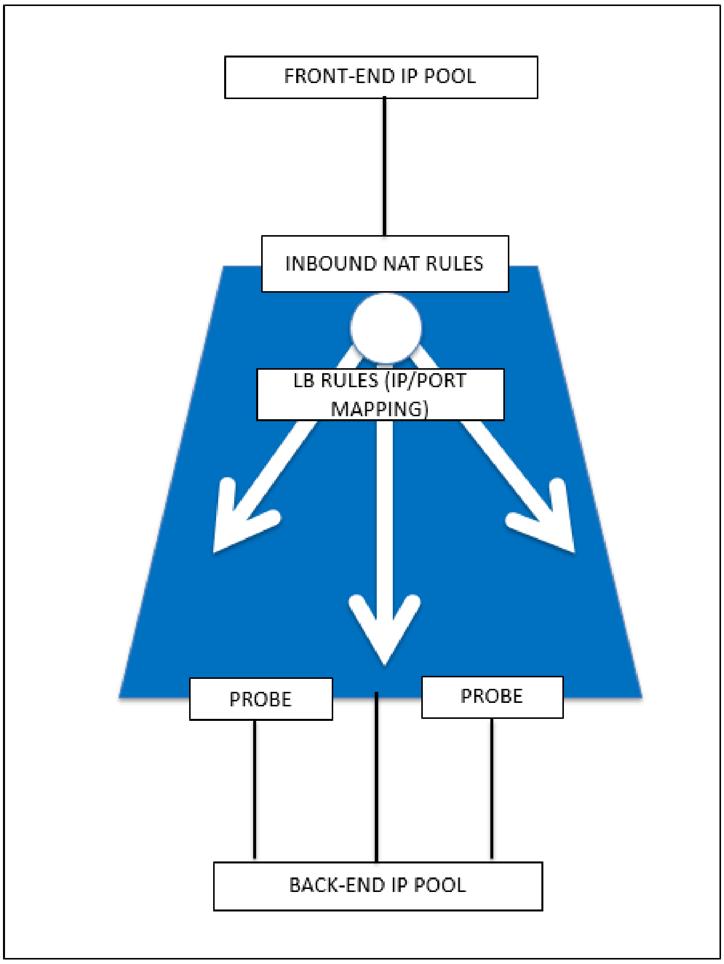
6. Instance Level Public IP (ILPIP) – An ILPIP is a public IP address that you can assign directly to your virtual machine or role instance, rather than to the cloud service that your virtual machine or role instance resides in. This does not take the place of the VIP (virtual IP) that is assigned to your cloud service. Rather, it's an additional IP address that you can use to connect directly to your virtual machine or role instance.

Note: In the past, an ILPIP was referred to as a PIP, which stands for public IP.

7. Inbound NAT Rules – This contains rules mapping a public port on the load balancer to a port for a specific virtual machine in the back end address pool.

8. IP-Config - It can be defined as an IP address pair (public IP and private IP) associated with an individual NIC. In an IP-Config, the public IP address can be NULL. Each NIC can have multiple IP-Config associated with it, which can be upto 255.

9. Load Balancing Rules – A rule property that maps a given front-end IP and port combination to a set of back-end IP addresses and port combination. With a single definition of a load balancer resource, you can define multiple load balancing rules, each rule reflecting a combination of a front end IP and port and back end IP and port associated with virtual machines.



10. Network Security Group (NSG) – NSG contains a list of Access Control List (ACL) rules that allow or deny network traffic to your virtual machine instances in a virtual network. NSGs can be associated with either subnets or individual virtual machine instances within that subnet. When a NSG is associated with a subnet, the ACL rules apply to all the virtual machine instances in that subnet. In addition, traffic to an individual virtual machine can be restricted further by associating a NSG directly to that virtual machine.

11. Private IP addresses – Used for communication within an Azure virtual network, and your on-premises network when you use a VPN gateway to extend your network to Azure. Private IP addresses allow Azure resources to communicate with other resources in a virtual network or an on-premises network through a VPN gateway or ExpressRoute circuit, without using an Internet-reachable IP address. In the Azure Resource Manager deployment model, a private IP address is associated with the following types of Azure resources – virtual machines, internal load balancers (ILBs), and application

gateways.

12. Probes – This contains health probes used to check availability of virtual machines instances in the back end address pool. If a particular virtual machine does not respond to health probes for some time, then it is taken out of traffic serving. Probes enable you to keep track of the health of virtual instances. If a health probe fails, the virtual instance will be taken out of rotation automatically.

13. Public IP Addresses (PIP) – PIP is used for communication with the Internet, including Azure public-facing services and is associated with virtual machines, Internet-facing load balancers, VPN gateways, and application gateways.

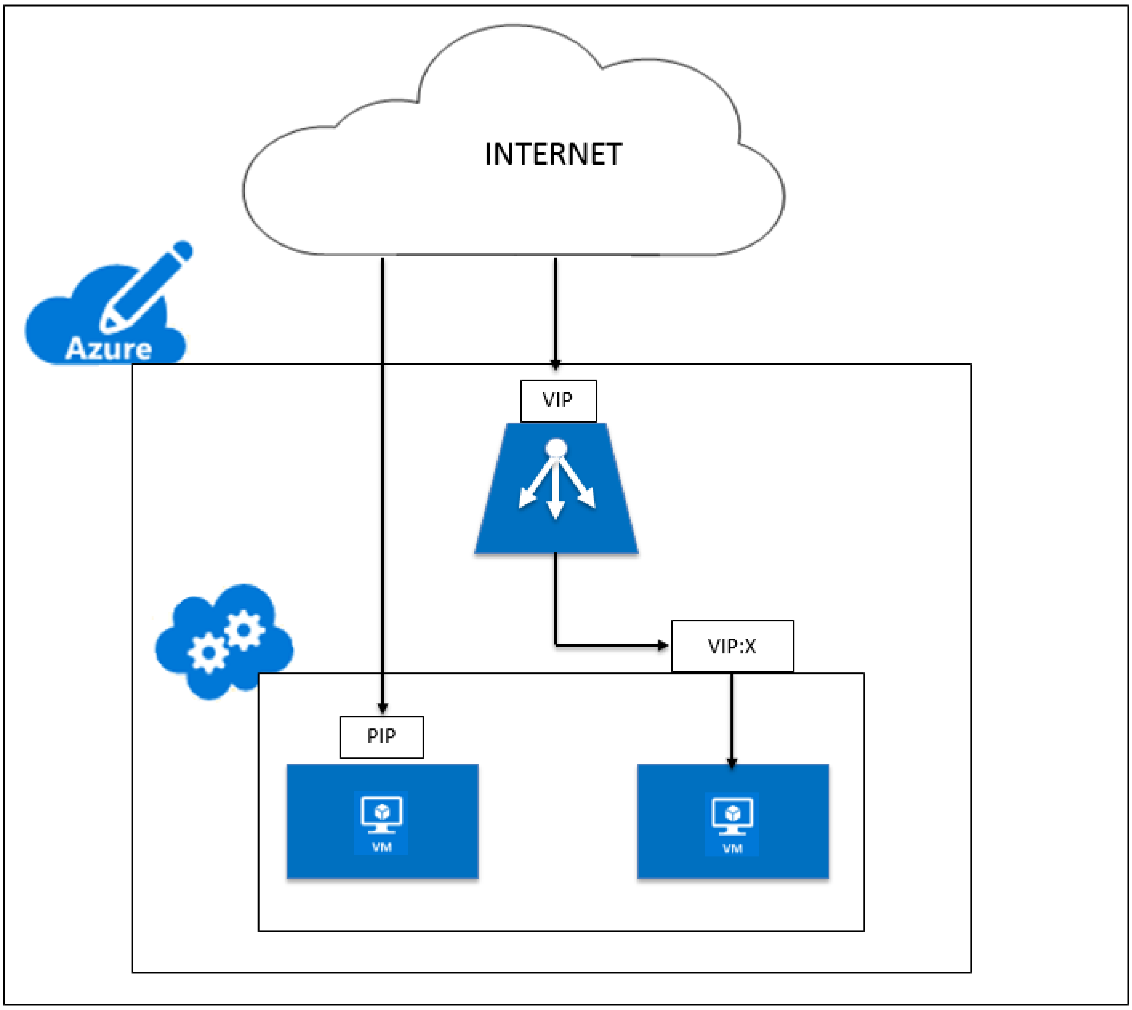
14. Region - An area within a geography that does not cross national borders and that contains one or more datacenters. Pricing, regional services, and offer types are exposed at the region level. A region is typically paired with another region, which can be up to several hundred miles away, to form a regional pair. Regional pairs can be used as a mechanism for disaster recovery and high availability scenarios. Also referred to generally as location.

15. Resource Group - A container in Resource Manager holds related resources for an application. The resource group can include all of the resources for an application, or only those resources that are logically grouped together

16. Storage Account – An Azure storage account gives you access to the Azure blob, queue, table, and file services in Azure Storage. Your storage account provides the unique namespace for your Azure storage data objects.

17. Virtual Machine – The software implementation of a physical computer that runs an operating system. Multiple virtual machines can run simultaneously on the same hardware. In Azure, virtual machines are available in a variety of sizes.

18. Virtual Network - An Azure virtual network is a representation of your own network in the cloud. It is a logical isolation of the Azure cloud dedicated to your subscription. You can fully control the IP address blocks, DNS settings, security policies, and route tables within this network. You can also further segment your VNet into subnets and launch Azure IaaS virtual machines and cloud services (PaaS role instances). Additionally, you can connect the virtual network to your on-premises network using one of the connectivity options available in Azure. In essence, you can expand your network to Azure, with complete control on IP address blocks with the benefit of enterprise scale Azure provides.



Deploy a NetScaler VPX instance on Oracle Public Cloud

September 21, 2020

> **Important**
>
> > NetScaler VPX instance on Oracle is in Limited Availability. If you'd like to participate, contact your local sales team or request a call at `https://www.citrix.com/products/netscaler-adc/form/inquiry/`.
>
> The NetScaler VPX virtual appliance (release 11.1 build 52.126) is available as a downloadable image with which you can create an instance in Oracle Public Cloud (OPC) environment. NetScaler VPX in OPC enables you to leverage cloud computing capabilities of OPC and use NetScaler load balancing and traffic management features for your business needs. You can deploy NetScaler

> VPX instances in OPC as standalone instances.

This section includes the following topics:

- Supported Features
- Hardware Requirements
- The Create Instance Wizard
- Terminology
- Limitations

Supported Features

A NetScaler VPX instance running in an Oracle Public Cloud deployment supports the following NetScaler features:

- Load Balancing
- ICA Proxy
- Content Switching
- AAA
- Rewrite
- Responder
- RDP Proxy
- nFactor
- LDAP
- VPN (CVPN/Full)
- GSLB

Hardware Requirements

VPX virtual appliances can be deployed in OPC as any instance type that has two or more cores and more than 2GB memory.

The Create Instance Wizard

You can create an instance on Oracle Public Cloud by using the **Create Instance** wizard. The wizard takes you through these pages. Use the forward and back button to move between pages or click the required page.

- Image: choose a template of a virtual hard disk of a specific size with an installed operating system. You can use images to create virtual machine instances in Oracle Compute Cloud Service.
- Shape: specify the OCPU and memory for your instance.
- Instance: enter details of your instance such as name, label, description, tags, customer attributes, and add SSH public key.

- Network: configure network settings for your instance such as IP networks, security lists, and so on.
- Storage: attach existing storage volumes, or create and attach a storage volume to the instance.
- Review: verify the settings for the new instance.

Terminology

This section describes the key terms that are used in this section.

- Instance: A virtual machine in Oracle Compute Cloud Service, created by using a specific machine image, with CPU and memory resources defined by a shape.
- IP Network: An IP subnet in your account. The address range of the IP network is determined by the IP address prefix that you specify while creating the IP network. These IP addresses aren't part of the common pool of Oracle-provided IP addresses used by the shared network. When you add an instance to an IP network, the instance is assigned an IP address in that subnet. You can assign IP addresses to instances either statically or dynamically, depending on your business needs. So you have complete control over the IP addresses assigned to your instances.
- Security Protocol: A protocol with which you can specify a transport protocol and the source and destination ports to use with the specified protocol. When you create a security rule, you can specify the security protocols that you want to use to permit traffic. Traffic is enabled by a security rule when the protocol in the packet matches the protocol specified here. If no protocol is specified, all protocols are allowed.
- Security Rule: A firewall rule that you can define to control network access to Oracle Compute Cloud Service instances through a specified security application.

For more information, see Oracle Compute Cloud Service Terminology.

Limitations

- You can't upgrade or downgrade a NetScaler VPX instance running on OPC.
- Only two-arm topology is supported.

Prerequisites

September 21, 2020

> **Important**
>
> NetScaler VPX instance on Oracle is in Limited Availability. If you'd like to participate, contact

> your local sales team or request a call at: https://www.citrix.com/products/netscaler-adc/form/inquiry/.

Before attempting to create a NetScaler instance in OPC, make sure you have the following:

- An Oracle Cloud Account
- An SSH key pair

Take the following steps to meet the above requirements:

1. Create and configure your account on Oracle Cloud.

   You must have an Oracle.com account to request a trial or purchase a subscription to an Oracle Cloud service. For more information about setting up your Oracle account, see Overview of Oracle Cloud Subscriptions.

2. Create an SSH key pair

   Before you create an instance in Oracle Public Cloud, you need to create and configure at least one SSH key pair, because after your instance is created, SSH key exchange is the first and only method of accessing the instance.

   a. Generate an SSH key pair by following the procedure described Generating an SSH Key Pair.

   Following is an example of creating an SSH key pair in RSA format.

   ssh-keygen -b 2048 -t rsa

   b. Upload the public key to the Oracle Public Cloud interface by following procedure described at Uploading the SSH Public Key. Once uploaded, you can assign the SSH key to any instance. You can also upload the key later while creating the instance.

Configure a NetScaler VPX standalone instance on Oracle Public Cloud

September 21, 2020

NetScaler VPX configuration on Oracle Public Cloud involves the following tasks:

1. Upload a NetScaler VPX Image and Associate It with the Oracle Cloud Web Console.
2. Configure the VPX instance.
3. Configure NetScaler owned IP Addresses and an LB virtual server.

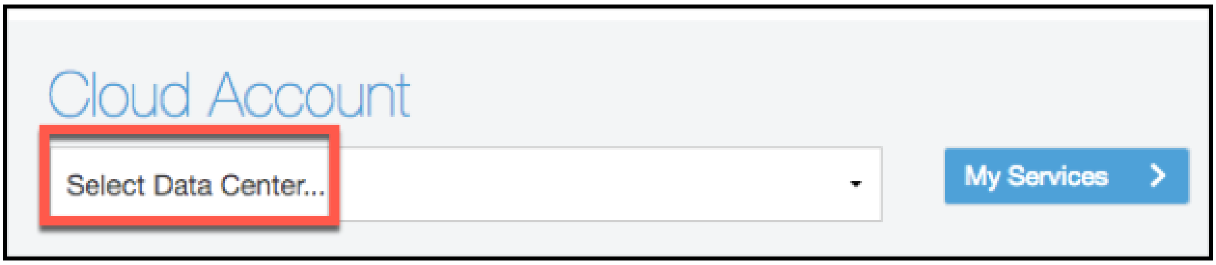
> **Note**
>
> Before you start configuring a NetScaler instance, see the Prerequisites.

Upload a NetScaler VPX image and associate it with the Oracle Cloud Web console

After obtaining the raw.tar.gz file image from Citrix, associate it with the Oracle Web console as follows:

1. Launch cloud.oracle.com and click **Sign In** in the upper-right corner.

2. From the **Cloud Account** drop-down menu, select the appropriate data center and click **My Services**.

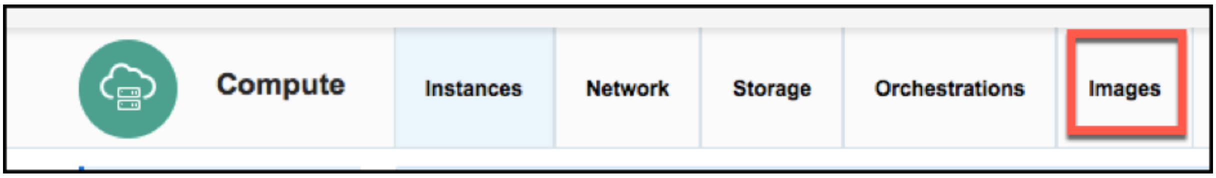

Cloud Account
Select Data Center...
My Services

3. In the **Enter your Identity Domain** field, enter the identity domain that Oracle assigned to your account, and click **Go.**

4. Enter your user name and password and click **Sign In**.

5. From the Dashboard, click **Compute > Open Service Console.**

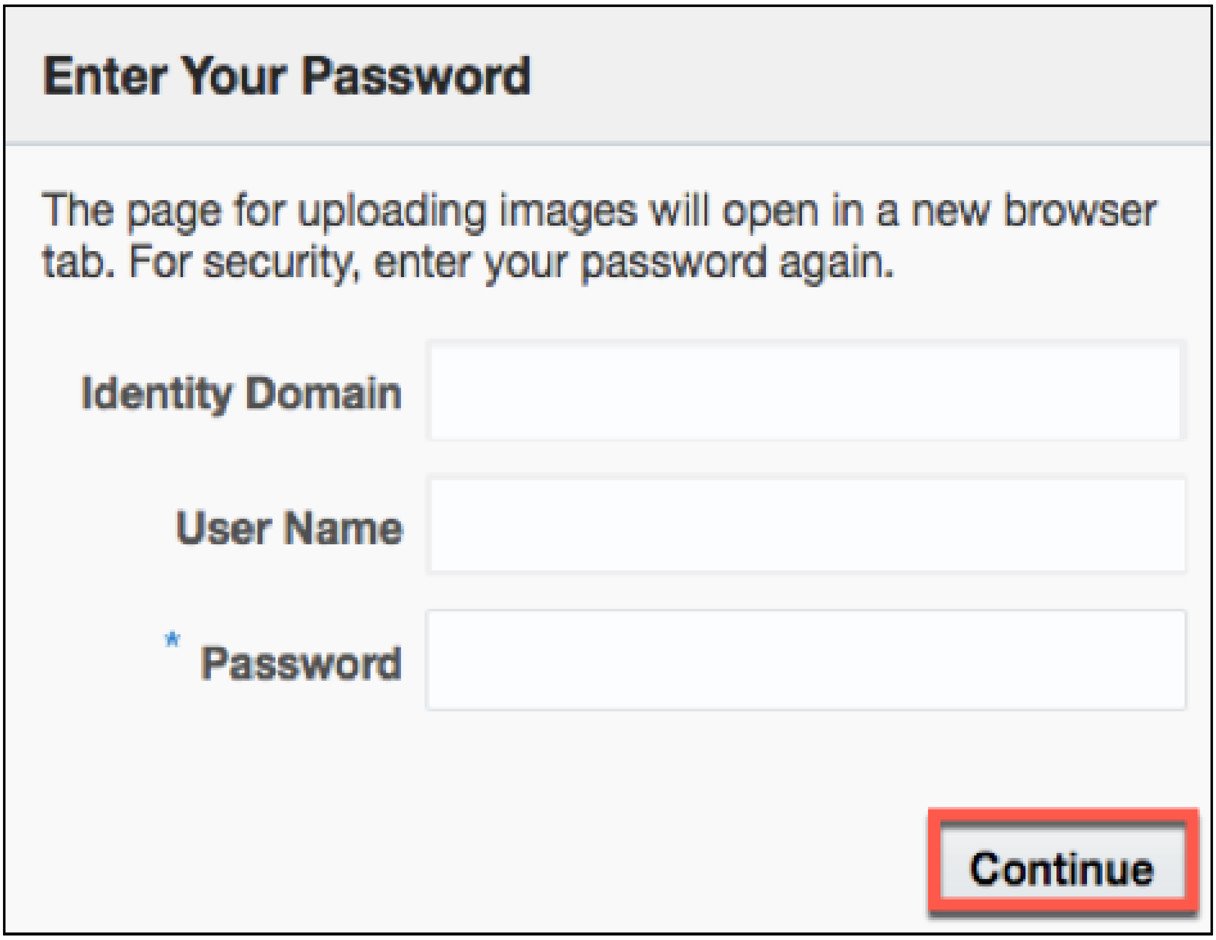
6. Click **Images.**


Compute
Instances
Network
Storage
Orchestrations
Images

7. The **Private Images** page appears.

8. Click **Upload Image**.

9. Enter your password, and then click **Continue**.

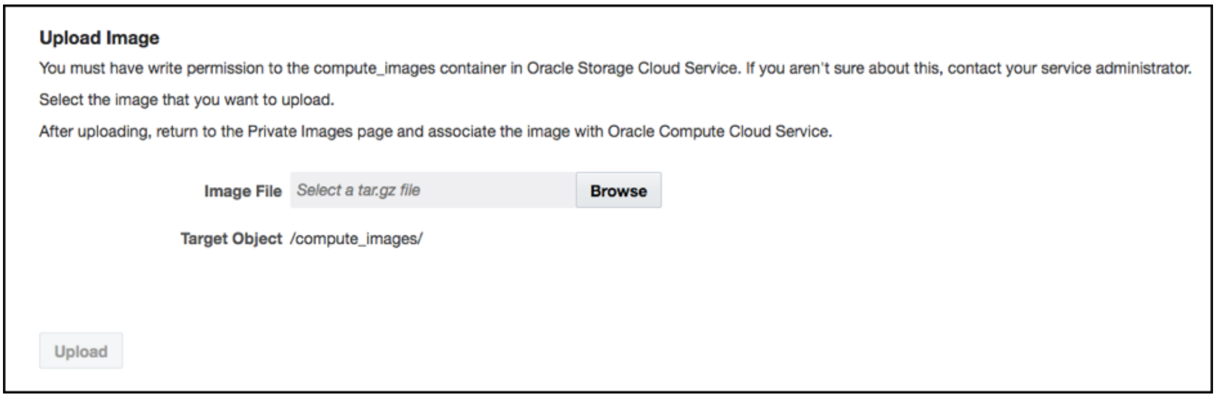


Enter Your Password

The page for uploading images will open in a new browser tab. For security, enter your password again.

10. The **Upload Image** page appears in a new tab.

11. In the **Image File** field, browse to select the raw.tar.gz image file that you want to upload.



Upload Image

You must have write permission to the compute_images container in Oracle Storage Cloud Service. If you aren't sure about this, contact your service administrator.

Select the image that you want to upload.

After uploading, return to the Private Images page and associate the image with Oracle Compute Cloud Service.

Target Object /compute_images/

12. In the **Target Object** field, enter the name of the object that the image file should be stored as in Oracle Storage Cloud Service.

13. Click **Upload**.

14. After the image is uploaded, it appears under **Private Images**.

15. Click the image and click **Associate Image**.
16. The **Associate Image** window appears.
17. Specify a name and description for the image that is being created.
18. Click **Select File** to select the file.
19. Click **OK** to associate the new image with the file.

Configure a NetScaler instance

Complete these steps to a VPX instance.

1. Launch the Create Instance wizard to create instances. Select and enter the details in the **Shape** and **Instance** pages. You can find more information about how to create an instance on the Oracle docs site.
2. In the **Network** page, clear the **Shared Network** check box and click **Configure Interface** > **Create IP Network** to create an IP network for the management IP (NIC 1/1) and click **Create**. Specify the other details in the **Configure IP Network Interface** window and click **Save**. Repeat the steps for NIC 1/2 and NIC 1/3. You can find more information about IP Network Options on the Oracle docs site.

   > **Note**
   >
   > In **Configure IP Network Interface**, select the **Default Gateway** check box.
   >
   > Set Public IP Address to **Auto Generated** for NIC 1/1 and 1/2 and specify private IP addresses for all three NICs.

3. On the **Storage** page, you can attach existing storage volumes to your instance, if necessary, or create storage volumes and attach them to the instance. In this example, select the default storage.
4. On the **Review** page, verify the information that you've entered, and then click **Create**.

   Monitor the status of the instance. When the status is shown as "Running," the instance is ready

5. Create an **IP Address Prefix Set**. You can find more information about how to create an IP address prefix set on the Oracle docs site.
6. Create **Security Protocols**. You can find more information about how to create a security protocol on the Oracle docs site.
7. Create **Security Rules**. You can find more information about how to create a security rule for IP networks on the Oracle docs site.

Configure NetScaler owned IP addresses and an LB virtual server

1. Configure the NetScaler owned IP addresses by using the NetScaler GUI or the command "add ns ip." For more information, see Configuring NetScaler-Owned IP Addresses.
2. Configure the NetScaler instance as a load balancing virtual server. For more information, see Set up basic load balancing.

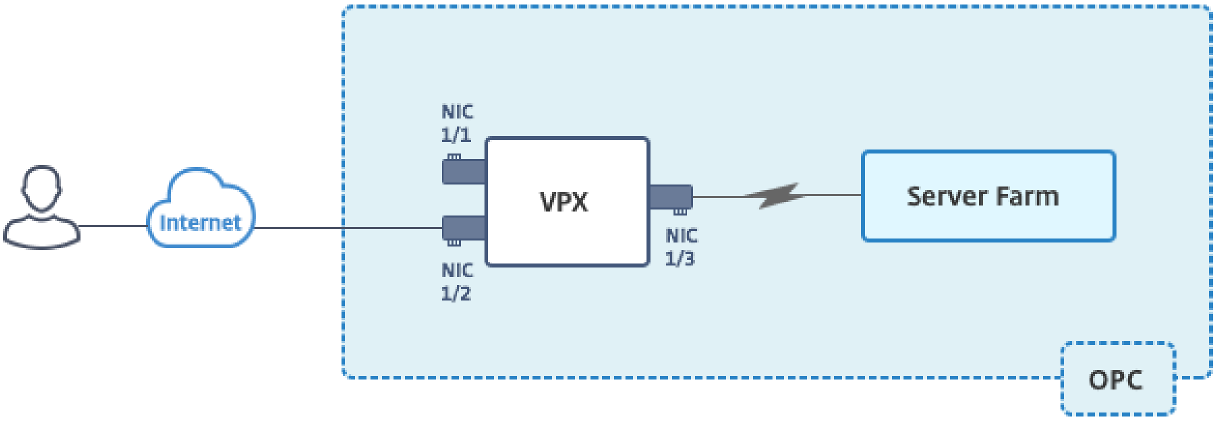
Scenario

September 21, 2020

> **Important**
>
> NetScaler VPX instance on Oracle is in Limited Availability. If you'd like to participate, contact your local sales team or request a call at `https://www.citrix.com/products/netscaler-adc/form/inquiry/`.

This scenario illustrates how to deploy a NetScaler VPX standalone instance in Oracle Public Cloud (OPC). The user creates a standalone VPX instance with multiple NICs. The instance, which is configured as a load balancing virtual server, communicates with back-end servers (the server farm). For this configuration, you have to set up the required communication routes between the instance and the back-end servers, and between the instance and the external hosts on the public Internet.

This scenario illustrates how to deploy a NetScaler VPX standalone instance in Oracle Public Cloud (OPC). As a user, create a standalone VPX instance with multiple NICs. The instance, which is configured as a load balancing virtual server, communicates with back-end servers (the server farm). For this configuration, you have to set up the required communication routes:



You create three NICs. Each NIC can be configured with a pair of IP addresses (public and private). The NICs serve the following purposes.

NIC	Purpose	Associated with
NIC 1/1	Serves management traffic (NSIP)	Public IP address and private IP address
NIC 1/2	Serves client-side traffic (VIP)	Public IP address and private IP address
NIC 1/3	Communicates with back-end servers (SNIP)	Private IP address(Public IP is not mandatory)

The following public IP addresses are used in this example.

Entity	Public IP address
NSIP	192.168.30.2
VIP	192.168.10.2
SNIP	192.168.20.2
Back-end server1	192.168.20.10
Back-end server2	192.168.20.11

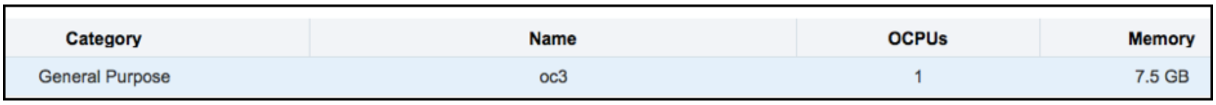
> **Note**
>
> Oracle assigns a static public IP address and a dynamic private IP address. For more information on Network Settings, see the Oracle docs site.

To create the required instances and set up the required security rules for this scenario, complete the following tasks:

1. From the Oracle Web Console, click **Create Instance** and click **Private Images**.
2. Select the required image and click **Review and Create**. The **Create Instance** wizard starts, displaying the default settings.

> **Note**
>
> If you click **Create** without going through the pages in the wizard, an image with the default settings is created. No SSH keys are associated with it. So make sure you enter the necessary details on each page of the wizard.
>
> While deploying a NetScaler VPX instance on Oracle Public Cloud, an SSH key pair is mandatory. However, the user cannot use the key pair to log on to the VPX instance. The user must use nsroot as the user name and as the password to log on.

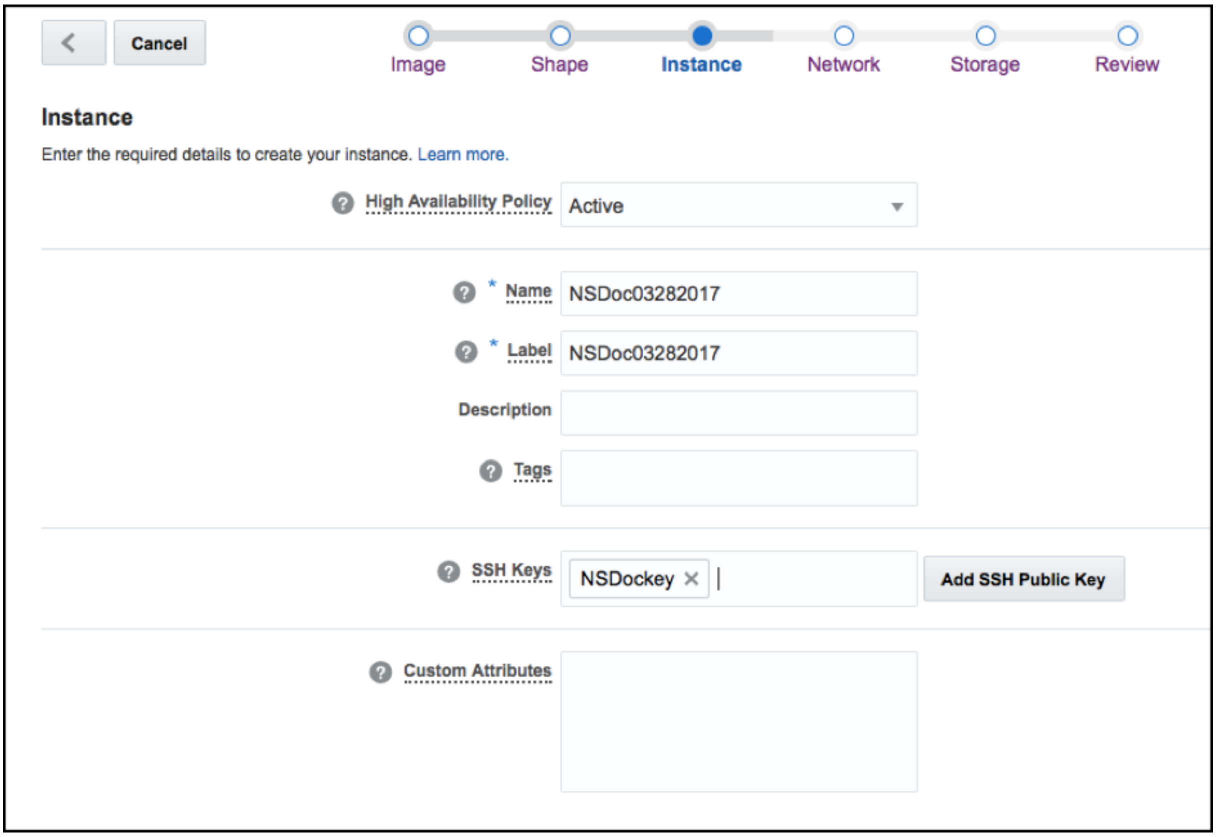
3. On the **Shape** page, select the shape that you want to use. The shape specifies the OCPU and memory resources to be allocated to the instance. Click the arrow next to the Review and Create tab on the upper right corner to go to the Instance page. In this scenario, you create an instance with 1 OCPU.



Category	Name	OCPUs	Memory
General Purpose	oc3	1	7.5 GB

4. On the Instance page, select or enter the following, and then go to the next page:

- High Availability Policy: Active
- `Name: \<Instance name\>`
- Label: default value
- Description: None
- Tags: None
- SSH Keys: Add the SSH key that you created in the Prerequisites section.
- Custom Attributes: None



Instance

Enter the required details to create your instance. Learn more.

High Availability Policy: Active

* Name: NSDoc03282017

* Label: NSDoc03282017

Description:

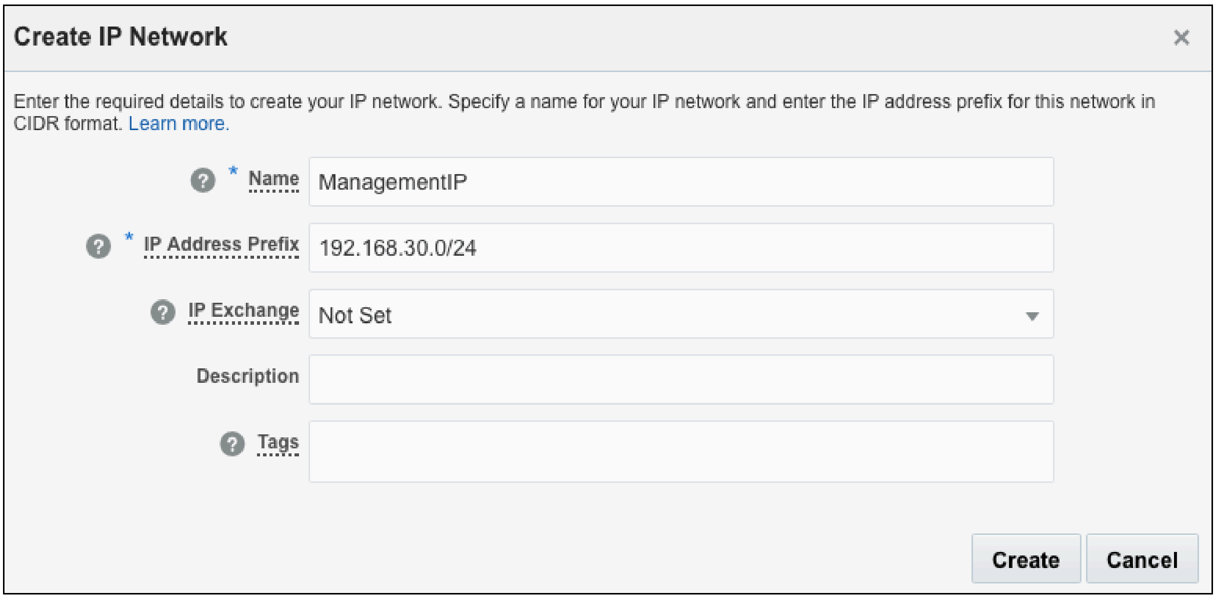
Tags:

SSH Keys: NSDockey ×

Custom Attributes:

5. On the **Network** page, clear the **Shared Network** check box and click **Configure Interface**.

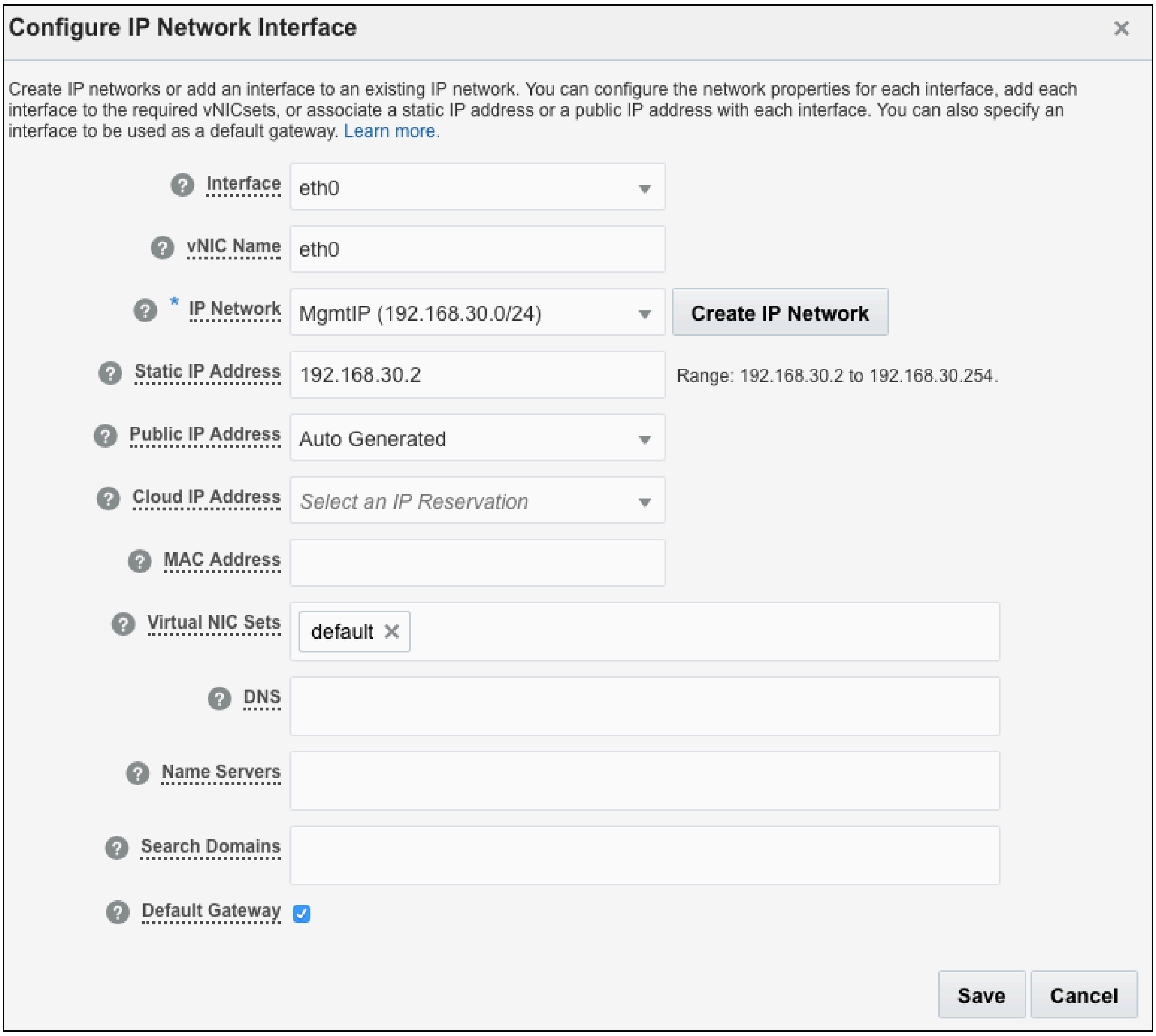
6. In the Configure IP Network Interface window, click **Create IP Network** to create an IP network for the management IP (NSIP). Add an IP Address Prefix (192.168.30.0/24), a name, and a description. Click **Create**



Enter the required details to create your IP network. Specify a name for your IP network and enter the IP address prefix for this network in CIDR format. Learn more.

* Name	ManagementIP
* IP Address Prefix	192.168.30.0/24
IP Exchange	Not Set
Description	
Tags	

7. In the Configure IP Network Interface window, select the IP Network that you've just created, add a static IP address, and select an auto-generated public IP address. Make sure the **Default Gateway** check box is selected.



Create IP networks or add an interface to an existing IP network. You can configure the network properties for each interface, add each interface to the required vNICsets, or associate a static IP address or a public IP address with each interface. You can also specify an interface to be used as a default gateway. Learn more.

Interface: eth0

vNIC Name: eth0

* IP Network: MgmtIP (192.168.30.0/24)

Static IP Address: 192.168.30.2

Range: 192.168.30.2 to 192.168.30.254.

Public IP Address: Auto Generated

Cloud IP Address: *Select an IP Reservation*

MAC Address:

Virtual NIC Sets: default ×

DNS:

Name Servers:

Search Domains:

Default Gateway: ☑

8. Similary create IP networks for NIC 1/2 and NIC 1/3 as follows.

NIC 1/2

- Name: IPConfig1 (for NIC 1/2)
- IP Address Prefix: 192/168.10.0/24

NIC 1/3

- Name: IPConfig2 (for NIC 1/3)
- IP Address Prefix: 192.168.20.0/24

9. Configure IP Network Interface as follows:

NIC 1/2

- Interface: eth1
- IP Network: IPConfig1
- Static IP Addresses: 192.168.10.2

- Public IP Address: Auto Generated

NIC 1/3

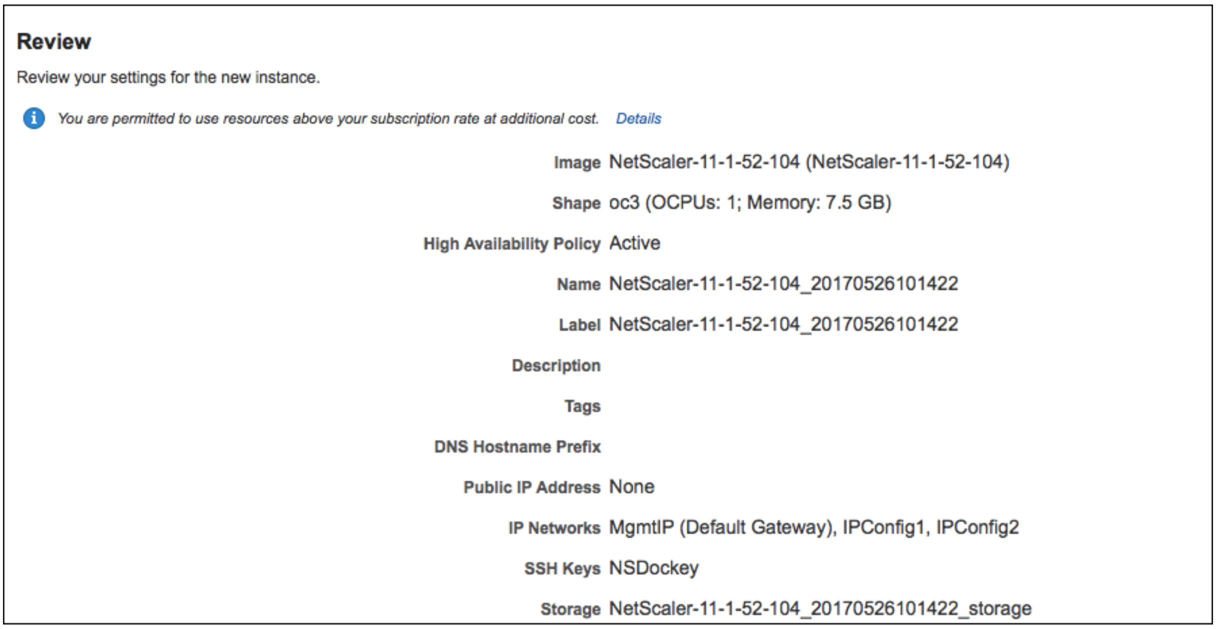
- Interface: eth2
- IP Network: IPConfig2
- Static IP Addresses: 192.168.20.2

Next, click the arrow to go to the **Storage** page.

> **Note**
>
> For NIC 1/2 and 1/3, do not select the **Default Gateway** check box.

8. On the Storage page, you can attach existing storage volumes to your instance, if necessary, or create storage volumes and attach them to the instance. In this example, select the default storage.

9. On the Review page, verify the information that you've entered, and then click **Create**.



Review

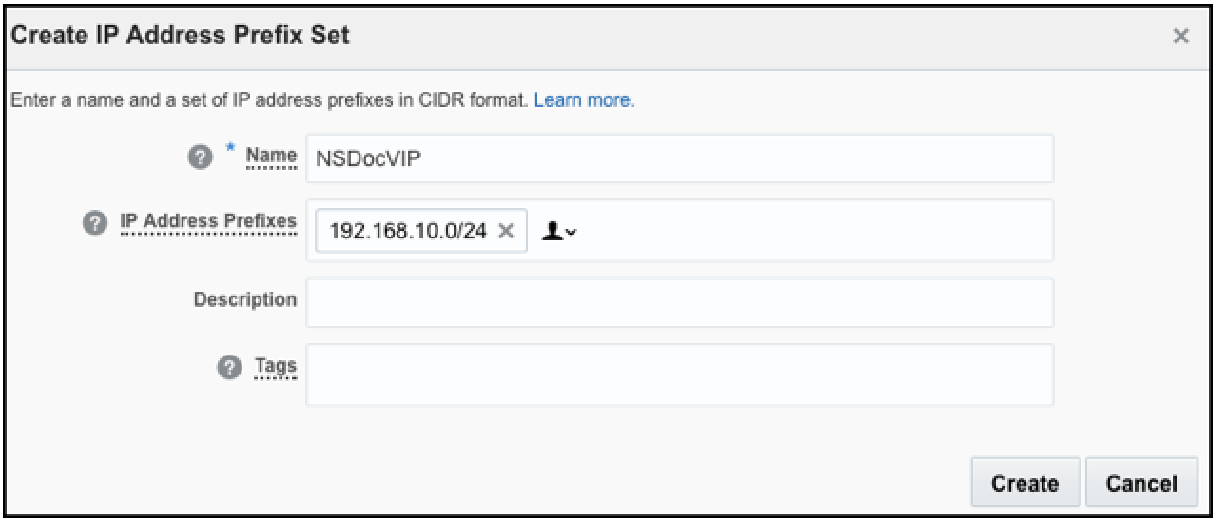
Review your settings for the new instance.

You are permitted to use resources above your subscription rate at additional cost. Details

Image	NetScaler-11-1-52-104 (NetScaler-11-1-52-104)
Shape	oc3 (OCPUs: 1; Memory: 7.5 GB)
High Availability Policy	Active
Name	NetScaler-11-1-52-104_20170526101422
Label	NetScaler-11-1-52-104_20170526101422
Description	
Tags	
DNS Hostname Prefix	
Public IP Address	None
IP Networks	MgmtIP (Default Gateway), IPConfig1, IPConfig2
SSH Keys	NSDockey
Storage	NetScaler-11-1-52-104_20170526101422_storage

10. Monitor the status of the instance. When the status is shown as "Running," the instance is ready. Follow the same steps to create two back-end servers.

11. From the Oracle web console, click **IP Network** > **IP Address Prefix Sets** > **Create IP Network**. Specify the name and the IP Address Prefix Set.

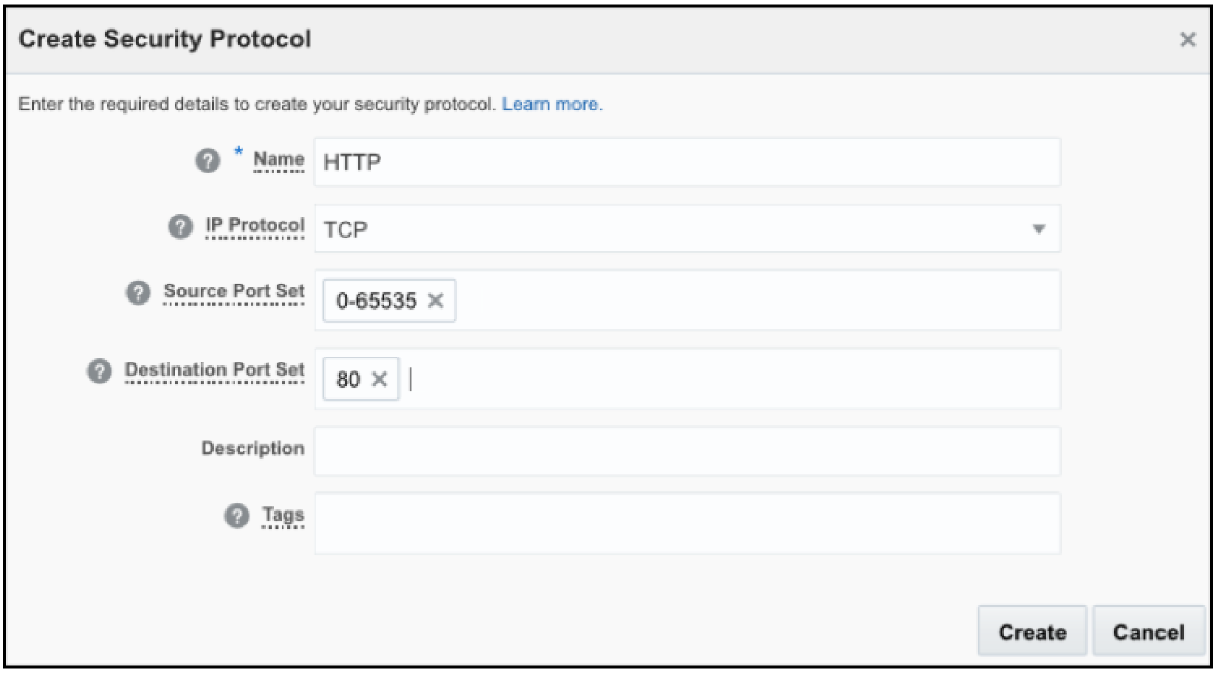


Create IP Address Prefix Set

Enter a name and a set of IP address prefixes in CIDR format. Learn more.

* Name	NSDocVIP
IP Address Prefixes	192.168.10.0/24
Description	
Tags	

12. Create a security protocol with which to create security rules. You'll create a security protocol for HTTP.

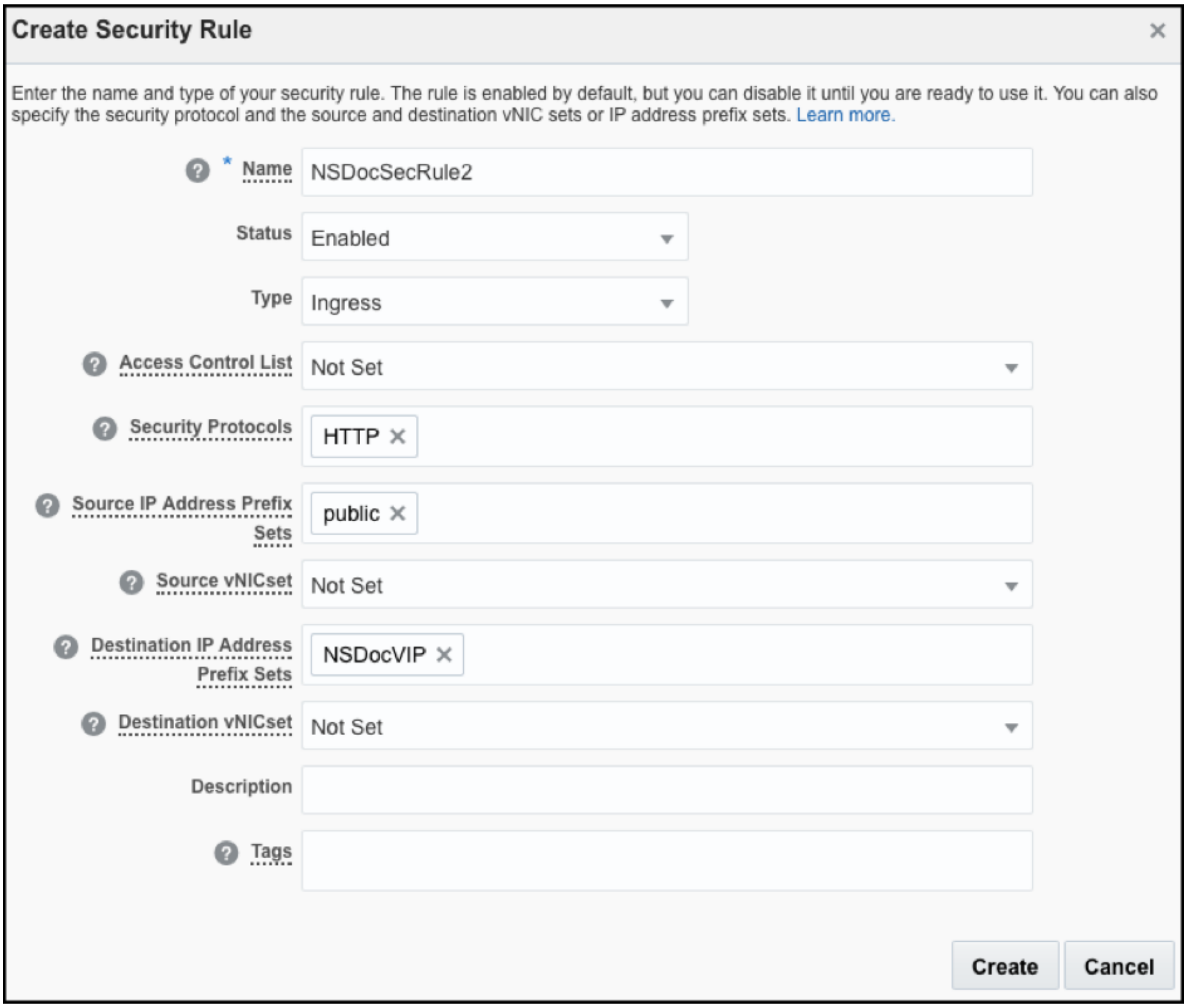


Create Security Protocol

Enter the required details to create your security protocol. Learn more.

* Name	HTTP
IP Protocol	TCP
Source Port Set	0-65535
Destination Port Set	80
Description	
Tags	

13. Create a security rule to allow external traffic to access the NetScaler Instance. For this scenario, create a rule to allow HTTP requests from external traffic to the NetScaler instance.



Enter the name and type of your security rule. The rule is enabled by default, but you can disable it until you are ready to use it. You can also specify the security protocol and the source and destination vNIC sets or IP address prefix sets. Learn more.

* Name	NSDocSecRule2
Status	Enabled
Type	Ingress
Access Control List	Not Set
Security Protocols	HTTP ×
Source IP Address Prefix Sets	public ×
Source vNICset	Not Set
Destination IP Address Prefix Sets	NSDocVIP ×
Destination vNICset	Not Set
Description	
Tags	

Now you can log on to your instance by using either GUI or SSH and complete the initial configuration. To find the oracle-assigned NetScaler management IP address, in the Oracle web console, click **Instances**. To the find the NetScaler management IP address pair, click the instance details icon for the instance that you created.

You can use SSH to log on to your instance as an nsroot user, by using the following command:

ssh -i ./<private key> nsroot@<ip address>

When prompted, type the password nsroot.

Next, configure the NetScaler-owned IP addresses and the NetScaler instance as a load balancing virtual server:

- Configure the NetScaler-owned IP addresses by using the NetScaler GUI or the command “add ns ip.” For more information, see https://docs.citrix.com/en-us/netscaler/11-1/networking/ip-addressing/configuring-netscaler-owned-ip-addresses.html
- Configure the NetScaler instance as a load balancing virtual server. For more information, see https://docs.citrix.com/en-us/netscaler/11-1/load-balancing/load-balancing-setup.html .

Example: Here's a sample LB configuration done by using the NetScaler CLI.

```
Add nsip 192.168.20.2 255.255.255.0

Add lb vs v1 http 192.168.10.2 80

Add service s[1-2] 192.168.20.[10-11] http 80

Bind lb vs v1 s[1-2]

Add vlan 10

Bind vlan 10 -Ifnum 1/3 -Ipaddress 192.168.20.2 255.255.255.0
```

The preceding configuration is based on the following assumptions:

Entity	Private IP address
VIP	192.168.10.2
SNIP	192.168.20.2
Back-end server1	192.168.20.10
Back-end server2	192.168.20.11

Deploy a NetScaler VPX Instance on Cisco CSP 2100

September 21, 2020

You can deploy an SR-IOV-enabled NetScaler VPX instance on Cisco Cloud Services Platform (CSP) 2100 to configure network functions virtualization (NFV) for your environment. The Cisco CSP 2100 is a turnkey, open, x86 Linux Kernel-based virtual machine (KVM) software and hardware platform for data center NFV.

This section describes how to deploy a NetScaler VPX instance on Cisco CSP 2100 and enable SR-IOV on the instance.

Points to Note

- Supported NetScaler version: From release 11.1 56.x and later
- Qualified on CSP version 2.2.3 build 32

Deploy a NetScaler VPX Instance on Cisco CSP 2100

Deploy a NetScaler VPX instance on Cisco CSP 2100 is a 2-step process:

1. Upload the NetScaler VPX qcow image.
2. Enable SR-IOV on the VPX instance.

Complete the following steps:

1. Download the NetScaler VPX 11.1 56.x qcow image from the Citrix download site.
2. Log on to Cisco CSP 2100 with your logon credentials.
3. From the dashboard, select **Configuration > Repository**.
4. From the Repository window, select the plus sign.
5. Select **Browse** to browse to the downloaded NetScaler VPX qcow image.
6. Select **Upload** to upload the VPX image to CSP 2100.

Next, provision the VPX instance and add SR-IOV interfaces to it.

1. From the CSP dashboard, select **Configuration** > **Services**
2. Select the plus sign. The **Create Service** template appears. Enter the following specifications:
    - Name
    - Target Host Name
    - Image Name: qcow image that you've already downloaded
    - Number of cores: minimum two
    - Disk Space: Minimum 20 GB
    - RAM: Minimum 2,048 GB
3. To add an SR-IOV interface, select the plus sign next to VNIC. The **VNIC Configuration** window appears.
4. For **Interface Type**, select **Passthrough**.
5. Specify the VLAN range.
6. For **Passthrough Mode**, select **SR-IOV**.
7. For **Network Name**, select an SR-IOV supported NIC. Click **Submit**.
8. Click **Deploy** to complement the process. It might take a moment for the NetScaler VPX instance to be deployed on Cisco CSP 2100.
9. Next, log on to the NetScaler VPX instance to complete the initial configuration.

Limitations

The following are CSP 2100-related limitations:

- If you power off the NetScaler VPX instance by using the CSP 2100 GUI , the NetScaler VPX configuration might get lost.
- In host VLAN mode, the VPX Niantic SR-IOV interfaces must be configured to 9k MTU size.
- Due to vCPU scheduling issues on the CSP 2100 platform, latency, transmit stall, and transmit queue overflow might be observed on the NetScaler VPX instance.
- The "Tag-all" command is not supported on a VPX Niantic SR-IOV interface.
- The LACP feature is not supported on a VPX Niantic SR-IOV interface.

Jumbo frames on NetScaler VPX instances

September 21, 2020

NetScaler VPX appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A NetScaler appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

For more information, seeConfiguring Jumbo Frames Support on a NetScaler Appliance.

Jumbo Frames support is available on NetScaler VPX appliances running on the following virtualization platforms:

- VMware ESX
- Linux-KVM Platform
- NetScaler XenServer
- Amazon Web Services (AWS)

Jumbo frames on VPX appliances works similar to Jumbo frames on MPX appliances. For more information on Jumbo Frames and its use cases, see Configuring Jumbo Frames on MPX appliances. The use cases of Jumbo Frames on MPX appliances also apply to VPX appliances.

Configure jumbo frames for a VPX instance running on VMware ESX

Perform the following tasks for configuring Jumbo Frames on a NetScaler VPX appliance running on VMware ESX server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501-9000. Use the CLI or GUI to set the MTU size. Note that NetScaler VPX appliances running on VMware ESX support receiving and transmitting jumbo frames containing up to only 9000 bytes of IP data.
2. Set the same MTU size on the corresponding physical interfaces of the VMware ESX server by using its management applications. For more information about setting the MTU size on the physical interfaces of VMware ESX, see http://vmware.com/.

Configure jumbo frames for a VPX instance running on Linux-KVM server

Perform the following tasks for configuring Jumbo Frames on a NetScaler VPX appliance running on a Linux-KVM Server:

1. Set the MTU of the interface or channel of the VPX appliance to a value in the range 1501-9216. Use the NetScaler VPX CLI or GUI to set the MTU size.
2. Set the same MTU size on the corresponding physical interfaces of a Linux-KVM Server by using its management applications. For more information about setting the MTU size on the physical interfaces of Linux-KVM, see http://www.linux-kvm.org/.

Configure jumbo frames for a VPX instance running on NetScaler XenServer

Perform the following tasks to configure jumbo frames on a NetScaler VPX appliance running on NetScaler XenServer:

1. Connect to the XenServer using XenCenter.
2. Shut down all the VPX instances that use the Networks for which the MTU must be changed.
3. On the **Networking** tab, select the network - network 0/1/2.
4. Select **Properties** and edit MTU.

After configuring jumbo frames on the XenServer, you can configure the jumbo frames on the NetScaler appliance. For more information, see Configuring Jumbo Frames Support on a NetScaler Appliance.

Configure jumbo frames for a VPX instance running on AWS

Host-level configuration is not required for VPX on Azure. To configure Jumbo Frames on VPX, follow the steps given in Configuring Jumbo Frames Support on a NetScaler Appliance.

Hardware Installation

September 21, 2020

> **Note**
>
> For the latest content, see NetScaler MPX.

The following sections describe the hardware installation and initial configuration for all NetScaler hardware platforms.

Hardware Platforms	Describes the NetScaler hardware platforms and provides detailed information about each platform and its components.
Preparing for Installation	Describes how to unpack the NetScaler appliance and prepare the site and rack for installing the appliance. Lists the cautions and warnings that you should review before you install the appliance.
Installing the Hardware	Describes the steps to install the rails, mount the hardware, connect the cables, and turn on the appliance.
Initial Configuration	Describes how to perform initial configuration of your NetScaler appliance and assign management and network IP addresses.
Lights Out Management Port of the NetScaler Appliance	Describes the different operations you can perform on your NetScaler appliance by using the Lights Out Management Port.

For information about NetScaler hardware and software compatibility and the supported upgrade and downgrade paths, see http://support.citrix.com/article/CTX113357.

Licensing

September 21, 2020

The following topics describe the migration instructions for setting up a new version of a NetScaler appliance with a list of all new and deprecated commands, parameters, and SNMP OIDs.

This document includes the following information:

- NetScaler Licensing Overview

NetScaler licensing overview

September 21, 2020

The process of allocating your NetScaler licenses has been greatly simplified. The new licensing framework allows you to focus on getting maximum value from Citrix products.

In the NetScaler GUI, you can use your hardware serial number (HSN) or your license access code to allocate your licenses. Alternatively, if a license is already present on your local computer, you can upload it to the appliance.

For all other functionality, such as returning or reallocating your license, you must use the licensing portal. Optionally, you can still use the licensing portal for license allocation. For more information about the licensing portal, see http://support.citrix.com/article/CTX131110.

Note: Purchase separate licenses for each appliance in a high availability pair. Ensure that the same types of licenses are installed on both the appliances. For example, if you purchase a platinum license for one appliance, you must purchase another platinum license for the other appliance.

Prerequisites

To use the hardware serial number or license access code to allocate your licenses:

- You must be able to access public domains through the appliance. For example, the appliance should be able to access www.citrix.com. The license allocation software internally accesses the Citrix licensing portal for your license. To access a public domain:
  - Use a proxy server or set up a DNS server.
  - Configure a NetScaler IP (NSIP) address or a subnet IP (SNIP) address on your NetScaler appliance.
- Your license must be linked to your hardware, or you must have a valid license access code. Citrix sends your license access code by email when you purchase a license.

Allocate a license by using the GUI

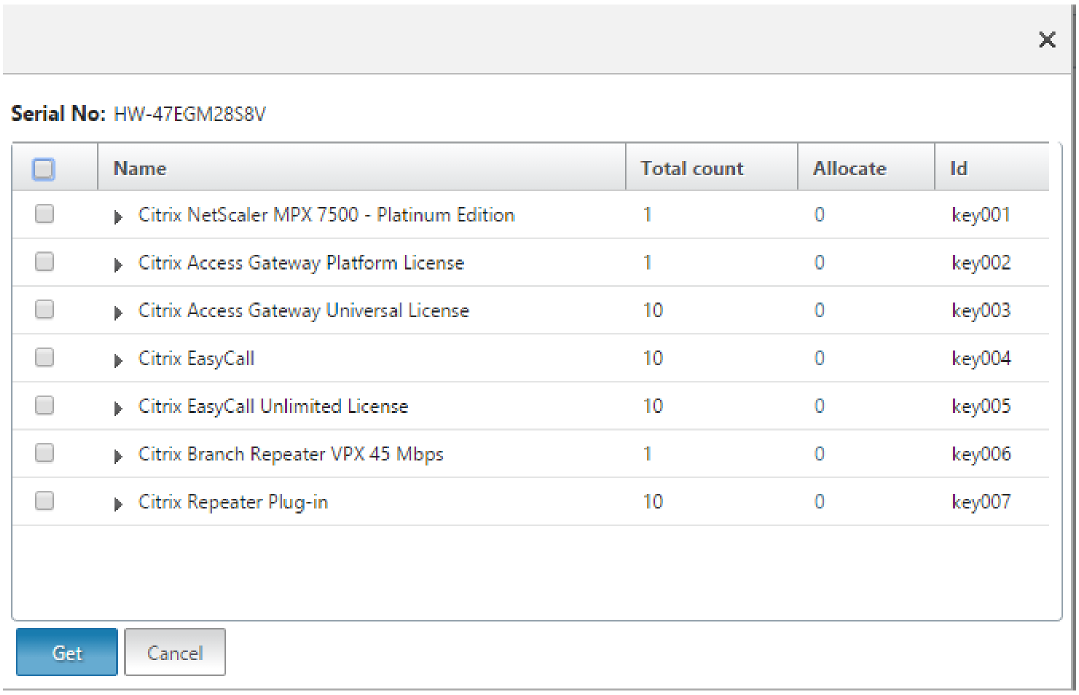
If your license is already linked to your hardware, the license allocation process can use the hardware serial number. Otherwise, you must type the license access code.

You can partially allocate licenses as required for your deployment. For example, if your license file contains 10 licenses, but your current requirement is for only six licenses, you can allocate six licenses now, and allocate more licenses later. You cannot allocate more than the total number of licenses present in your license file.

To allocate your license

1. In a web browser, type the IP address of the NetScaler appliance (for example, `http://192.168.100.1`).
2. In User Name and Password, type the administrator credentials.
3. On the **Configuration** tab, navigate to **System \ > Licenses**.
4. In the details pane, click **Manage Licenses**, click **Add New License**, and then select one of the following options:
    - **Use Serial Number.** The software internally fetches the serial number of your appliance and uses this number to display your licenses.
    - **Use license access code.** Citrix emails the license access code for the license that you purchased. Enter the license access code in the text box.

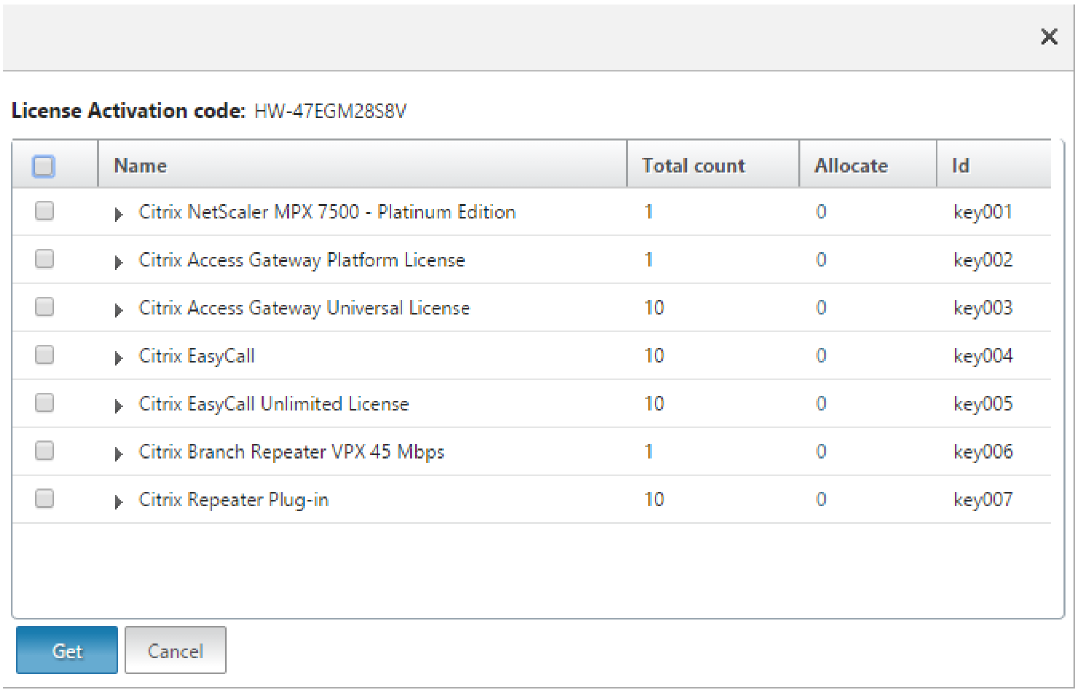
      If you do not want to configure internet connectivity on the NetScaler appliance, you can use a proxy server. Select the **Connect through Proxy Server** check box and specify the IP address and port of your proxy server.
5. Click **Get Licenses**. Depending on the option that you selected, one of the following dialog boxes appears.
    - The following dialog box appears if you selected Hardware Serial Number.



Serial No: HW-47EGM28S8V

☐	Name	Total count	Allocate	Id
☐	Citrix NetScaler MPX 7500 - Platinum Edition	1	0	key001
☐	Citrix Access Gateway Platform License	1	0	key002
☐	Citrix Access Gateway Universal License	10	0	key003
☐	Citrix EasyCall	10	0	key004
☐	Citrix EasyCall Unlimited License	10	0	key005
☐	Citrix Branch Repeater VPX 45 Mbps	1	0	key006
☐	Citrix Repeater Plug-in	10	0	key007

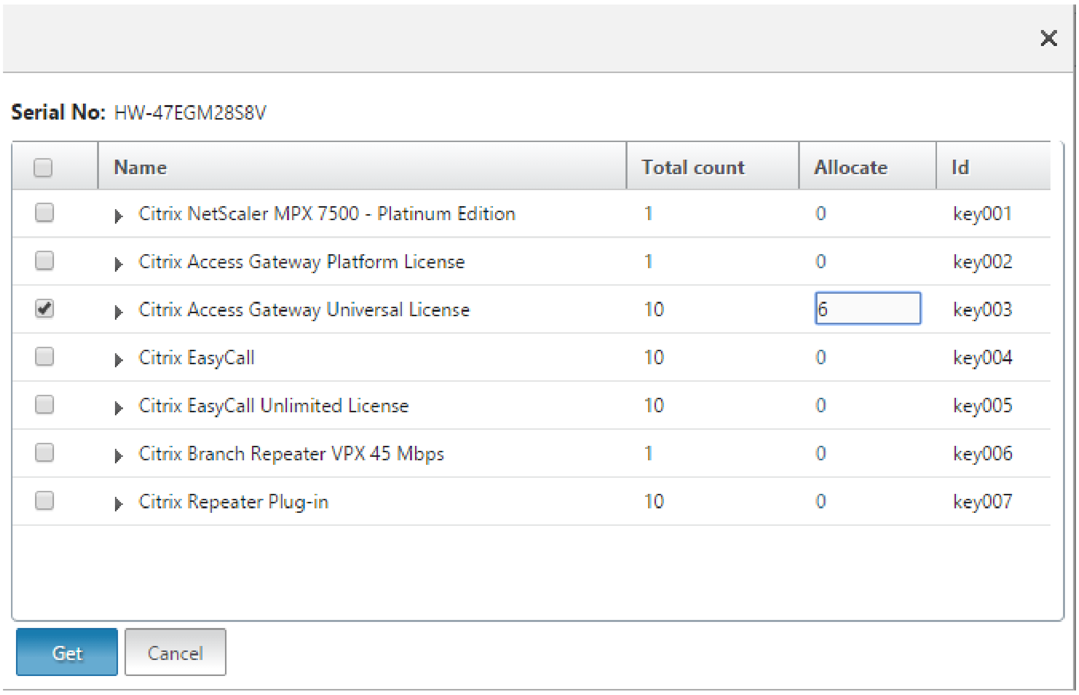
- The following dialog box appears if you selected license access code.



License Activation code: HW-47EGM28S8V

☐	Name	Total count	Allocate	Id
☐	Citrix NetScaler MPX 7500 - Platinum Edition	1	0	key001
☐	Citrix Access Gateway Platform License	1	0	key002
☐	Citrix Access Gateway Universal License	10	0	key003
☐	Citrix EasyCall	10	0	key004
☐	Citrix EasyCall Unlimited License	10	0	key005
☐	Citrix Branch Repeater VPX 45 Mbps	1	0	key006
☐	Citrix Repeater Plug-in	10	0	key007

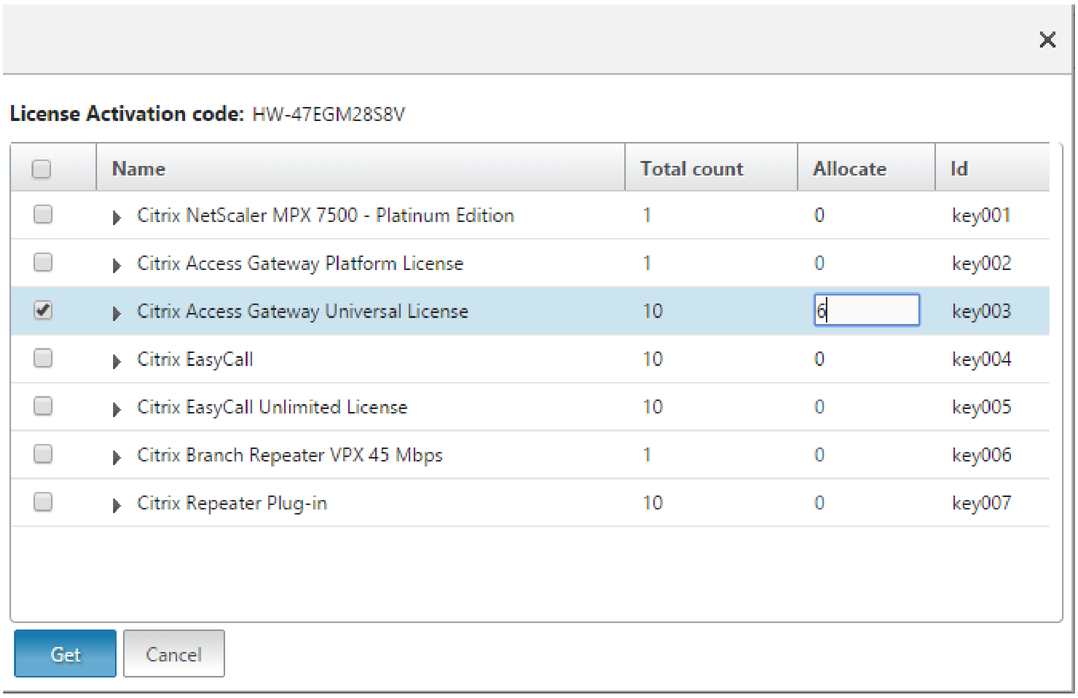
6. Select the license file that you want to use to allocate your licenses.
7. In the **Allocate** column, enter the number of licenses to be allocated. Then click **Get**.
    - If you selected **Hardware Serial Number**, enter the number of licenses, as shown in the following screenshot.



Serial No: HW-47EGM28S8V

☐	Name	Total count	Allocate	Id
☐	Citrix NetScaler MPX 7500 - Platinum Edition	1	0	key001
☐	Citrix Access Gateway Platform License	1	0	key002
☑	Citrix Access Gateway Universal License	10	6	key003
☐	Citrix EasyCall	10	0	key004
☐	Citrix EasyCall Unlimited License	10	0	key005
☐	Citrix Branch Repeater VPX 45 Mbps	1	0	key006
☐	Citrix Repeater Plug-in	10	0	key007

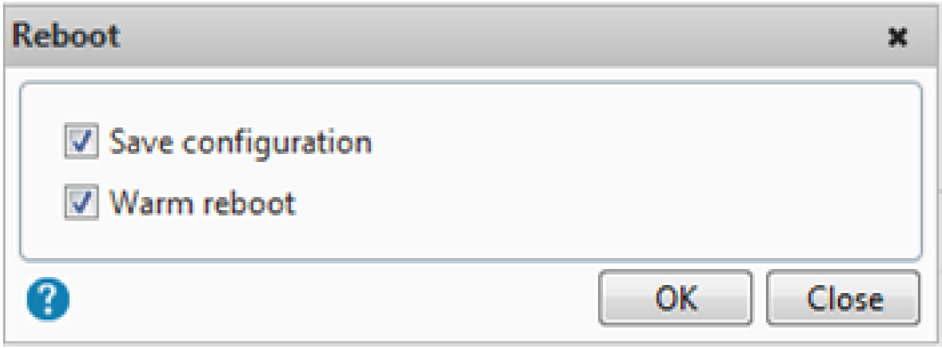
- If you selected **license access code**, enter the number of licenses, as shown in the following screenshot.



License Activation code: HW-47EGM28S8V

☐	Name	Total count	Allocate	Id
☐	Citrix NetScaler MPX 7500 - Platinum Edition	1	0	key001
☐	Citrix Access Gateway Platform License	1	0	key002
☑	Citrix Access Gateway Universal License	10	6	key003
☐	Citrix EasyCall	10	0	key004
☐	Citrix EasyCall Unlimited License	10	0	key005
☐	Citrix Branch Repeater VPX 45 Mbps	1	0	key006
☐	Citrix Repeater Plug-in	10	0	key007

8. Click restart for the license to take effect.
9. In the restart dialog box, click **OK** to proceed with the changes, or click **Close** to cancel the changes.



☑ Save configuration
☑ Warm reboot

Install a license

If you downloaded your license file to your local computer by accessing the licensing portal, you must upload the license to the appliance.

To install a license file by using the GUI

1. In a web browser, type the IP address of the NetScaler appliance (for example, `http://192.168.100.1`).
2. In User Name and Password, type the administrator credentials.
3. On the **Configuration** tab, navigate to System Licenses.
4. In the details pane, click **Manage Licenses**.
5. Click **Add New License**, then select **Upload license files from a local computer**.
6. Click **Browse**. Navigate to the location of the license files, select the license file, and then click **Open**.
7. Click restart to apply the license.
8. In the restart dialog box, click **OK** to proceed with the changes, or click **Close** to cancel the changes.

To install the licenses by using the CLI:

1. Open an **SSH connection** to the ADC appliance by using an SSH client, such as PuTTY.

2. Log on to the ADC appliance by using the administrator credentials.

3. Switch to the shell prompt, create a license subdirectory in the nsconfig directory, if it does not exist, and copy one or more new license files to this directory.

   Example

   ```
   login: nsroot
   Password: nsroot
   Last login: Mon Aug  4 03:37:27 2008 from 10.102.29.9
   Done
   > shell
   ```

```
Last login: Mon Aug  4 03:51:42 from 10.103.25.64
root@ns# mkdir /nsconfig/license
root@ns# cd /nsconfig/license
<!--NeedCopy-->
```

Copy one or more new license files to this directory.

Note: The NetScaler appliance does not prompt for a reboot option when you use the command line interface to install the licenses. Run the
reboot -w command to warm restart the system, or run the
restart command to restart the system normally.

Verify licensed features

Before using a feature, ensure that your license supports the feature.

To verify the licensed features by using the CLI

1. Open an **SSH connection** to the ADC appliance by using an SSH client, such as PuTTY.
2. Log on to the ADC appliance by using the administrator credentials.
3. At the command prompt, enter the sh ns license command to display the features supported by the license.

   Example

   ```
   sh ns license
           License status:
                              Web Logging: YES
                         Surge Protection: YES
                           .......

                           HTML Injection: YES
    Done
   <!--NeedCopy-->
   ```

To verify the licensed features by using the GUI

1. In a web browser, type the IP address of the ADC appliance, such as `http://192.168.100.1`.
2. In User Name and Password, type the administrator credentials.
3. Provide the User name and Password and click **Login**.
4. In the navigation pane, expand **System**, and then click **Licenses**. You see a green check mark next to the licensed features.

Enable or disable a feature

When you use the NetScaler appliance for the first time, you must enable a feature before you can use its functionality. If you configure a feature before it is enabled, a warning message appears. The configuration is saved but it applies only after the feature is enabled.

To enable a feature by using the CLI

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

- enable feature <FeatureName>
- show feature

  Example

  ```
  enable feature lb cs
  done
   >show feature

          Feature                        Acronym                Status
          -------                        -------                ------
   1)     Web Logging                    WL                     OFF
   2)     Surge Protection               SP                     ON
   3)     Load Balancing                 LB                     ON
   4)     Content Switching              CS                     ON
   5)     Cache Redirection              CR                     ON
   .
   .
   .
   24)    NetScaler Push                 push                   OFF
   Done
  <!--NeedCopy-->
  ```

  The example shows how to enable load balancing (lb) and content switching (cs).

  If the license key is not available for a particular feature, the following error message appears for that feature:

  ERROR: feature(s) not licensed

  Note: To enable an optional feature, you must have a feature-specific license. For example, you have purchased and installed the Citrix NetScaler Enterprise Edition license. However, to enable the Integrated Caching feature, you must purchase and install the AppCache license.

To disable a feature by using the CLI

At the command prompt, type the following commands to disable a feature and verify the configuration:

- disable feature <FeatureName>
- show feature

  Example

  The following example shows how to disable load balancing (LB).

  ```
  > disable feature lb
   Done
  > show feature

          Feature                        Acronym              Status
          -------                        -------              ------
   1)     Web Logging                    WL                   OFF
   2)     Surge Protection               SP                   ON
   3)     Load Balancing                 LB                   OFF
   4)     Content Switching              CS                   ON
   .
   .
   .
   24)    NetScaler Push                 push                 OFF
   Done
  >
  <!--NeedCopy-->
  ```

NetScaler Pooled Capacity

September 21, 2020

NetScaler pooled capacity is a licensing framework that comprises a common bandwidth and instance pool that is hosted on and served by Citrix Application Delivery Management.

For complete information, see NetScaler Pooled Capacity "NetScaler Pooled Capacity".

Upgrade and downgrade a NetScaler appliance

September 21, 2020

NetScaler 11.1 offers new and updated features with increased functionality. A comprehensive list of enhancements is documented in the release notes accompanying the release announcement. Read the release notes before you upgrade your software.

This section provides information about upgrading and downgrading a NetScaler appliance (MPX and VPX) firmware. For information about upgrading a NetScaler SDX appliance, see Single Bundle Upgrade.

Note: For upgrading or downgrading the nodes in a cluster setup, see Upgrading or Downgrading the Cluster Software.

Upgrading from release 10.1 build 121.10 or any earlier releases to release 10.1 build 122.17 and later involves some location changes of user monitor script files. For details, see the **Directory locations of script files for user monitors** section in Upgrade a Citrix NetScaler standalone appliance.

> **Note**
>
> For best practices for upgrading NetScaler appliances, read the article CTX126793.

Before you begin

September 21, 2020

Before you start the upgrade or downgrade process, make sure you check the following:

- The licensing framework and types of licenses. A software edition upgrade might require new licenses, such as upgrading from the standard edition to the enterprise edition, the standard edition to the platinum edition, or the enterprise edition to the platinum edition. Existing Citrix NetScaler licenses continue to work when you upgrade to version 11.1. For more information, see Licensing
- Check for New and deprecated commands, parameters, and SNMP OIDs.
- Check forCitrix NetScaler Hardware and Software Compatibility Matrix.
- If the Citrix NetScaler Gateway logon page is customized, then make sure that the UI theme is set to default.
- Review the LOM Firmware Upgrade page, if you are upgrading LOM.

- Download the Citrix NetScaler firmware from the download page. For the detailed steps to download the Citrix NetScaler firmware, see the "How to download and update the NetScaler Firmware" section in this article CTX205476.
- Perform a back up of the configuration file, customization file, certificates, monitor scripts, license files, and so on either manually or refer to the following documentation for back up using Citrix NetScaler CLI or GUI - Backup and restore.
- Validate the integrity of the Citrix NetScaler and check available space before performing an upgrade. For more information about how to free up storage space, see https://support.citrix.com/article/CTX133588
- Check the Citrix ADC VPX Support matrix and usage guidelines.
- Check the FAQ section.

For more information about prerequisites for upgrading or downgrading the Citrix NetScaler firmware, see these support articles:

- CTX126793: Best Practices for Upgrading Citrix NetScaler or Citrix NetScaler Gateway Appliances
- CTX220371: Must Read Articles Before and After Upgrading Citrix NetScaler

Upgrade considerations - SNMP configuration

September 21, 2020

The timeout parameter for an SNMP alarm is an internal option that has no impact on the alarm configuration.

Timeout parameter might appear in the SNMP alarm configurations in the running configuration (sh running) and the saved configuration (ns.conf) even if you have not made any changes to these SNMP alarm configurations.

On upgrading to a release build with the fix for the timeout setting issue, the SNMP configurations are erroneously reset to default values.

The following SNMP alarms (if configured) are impacted during an upgrade:

- APPFW-BUFFER-OVERFLOW
- APPFW-COOKIE
- APPFW-CSRF-TAG
- APPFW-DENY-URL
- APPFW-FIELD-CONSISTENCY
- APPFW-FIELD-FORMAT

- APPFW-POLICY-HIT
- APPFW-REFERER-HEADER
- APPFW-SAFE-COMMERCE
- APPFW-SAFE-OBJECT
- APPFW-SQL
- APPFW-START-URL
- APPFW-VIOLATIONS-TYPE
- APPFW-XML-ATTACHMENT
- APPFW-XML-DOS
- APPFW-XML-SCHEMA-COMPILE
- APPFW-XML-SOAP-FAULT
- APPFW-XML-SQL
- APPFW-XML-VALIDATION
- APPFW-XML-WSI
- APPFW-XML-XSS
- APPFW-XSS
- CLUSTER-BACKPLANE-HB-MISSING
- CLUSTER-NODE-HEALTH
- CLUSTER-NODE-QUORUM
- CLUSTER-VERSION-MISMATCH
- COMPACT-FLASH-ERRORS
- CONFIG-CHANGE
- CONFIG-SAVE
- HA-BAD-SECONDARY-STATE
- HA-NO-HEARTBEATS
- HA-SYNC-FAILURE
- HA-VERSION-MISMATCH
- HARD-DISK-DRIVE-ERRORS
- HA-STATE-CHANGE
- HA-STICKY-PRIMARY
- PORT-ALLOC-FAILED
- SYNFLOOD

These SNMP alarms configurations are impacted when you upgrade the Citrix ADC (NetScaler) appliance to the following release builds:

- Release 11.1 build 61.2 or later
- Release 12.0 build 61.0 or later
- Release 12.1 build 30.1 or later
- Release 13.0 build 51.4 or later

Example

Let’s consider an example of CLUSTER-NODE-HEALTH SNMP alarm.

```
CLUSTER-NODE-HEALTH SNMP alarm is set up by using the Citrix ADC (NetScaler) command line:

> set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity Major

> save config
<!--NeedCopy-->
```

This SNMP alarm configuration appears in the saved configuration file (`ns.conf`) as:

```
set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout 86400

<!--NeedCopy-->
```

During an upgrade to any of the release builds mentioned above, the following error appears in ns.log file:

```
May 23 09:14:46 <local0.err> ns nsconfigd: __init_config_filter(): (null) line 0: No such argument [-timeout]>> set snmp alarm CLUSTER-NODE-HEALTH -time 111 -state DISABLED -severity Major -timeout 86400.
<!--NeedCopy-->
```

After the upgrade, the SNMP alarm configurations are reset to the default values.

Workaround

Use one of the following workarounds:

- Before the upgrade, remove the timeout setting from the SNMP configurations in the saved configuration file (ns.conf).
- After the upgrade, reconfigure the SNMP alarms without the timeout parameter.

Upgrade a Citrix NetScaler standalone appliance

September 21, 2020

Before upgrading the system software, ensure that you read the Before you begin section. Also, complete the prerequisites such as backing up the necessary files and downloading the Citrix NetScaler firmware.

When upgrading from 10.0 and higher releases, you can use the GUI or the CLI.

You cannot upgrade to release 11.1 from the following builds by using the GUI Upgrade Wizard:

- All builds of release 10.1
- Any build before Build 57.x of release 10.5

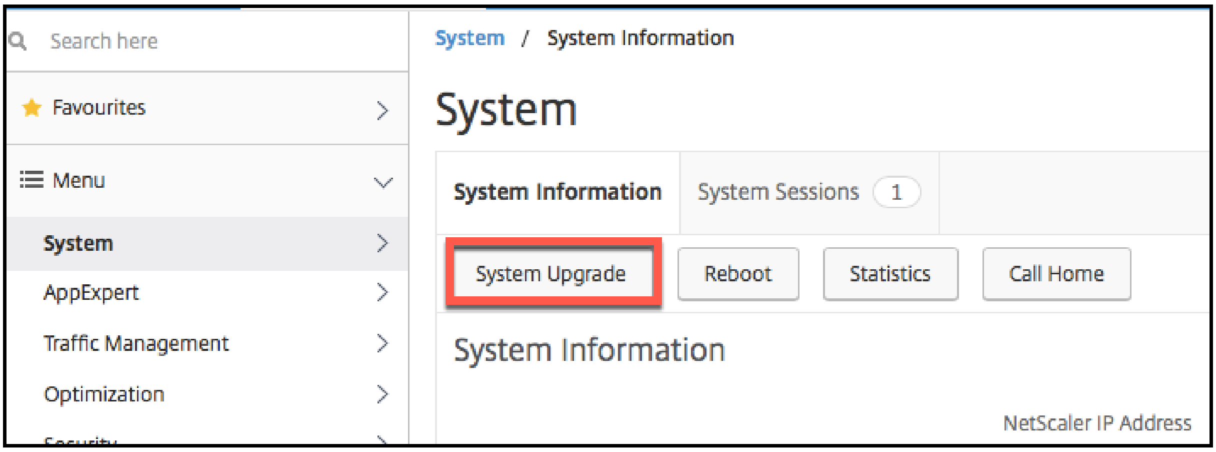
If your Citrix NetScaler runs any 9.x or lower release, visit the Product Matrix site for more information.

.

Upgrade a Citrix NetScaler standalone appliance by using the GUI

Follow these steps to upgrade a standalone Citrix NetScaler instance by using the GUI.

1. In a web browser, type the IP address of the Citrix NetScaler, for example `http://10.102.29.50`.
2. In User Name and Password, type the administrator credentials (nsroot/nsroot) and then click **Log On**.
3. From the GUI, click **System Upgrade**.


Search here
Favourites
Menu
System
AppExpert
Traffic Management
Optimization
Security
System / System Information
System
System Information
System Sessions 1
System Upgrade
Reboot
Statistics
Call Home
System Information
NetScaler IP Address

4. From the **Choose File** menu choose the appropriate option: **Local** or **Appliance**. If you want to use the Appliance option, the firmware needs to be uploaded to the Citrix NetScaler first. You can use any file transfer method such as WinSCP to upload the Citrix NetScaler firmware to the appliance.

5. Select the correct file and click **Upgrade**.

6. Follow the instructions to upgrade the software.

7. When prompted, select **Reboot**.

After the upgrade, close all browser instances and clear your computer's cache before accessing the appliance.

Upgrade a Citrix NetScaler standalone appliance by using the CLI

Follow these steps to upgrade a standalone Citrix NetScaler instance by using the CLI:

In the following procedure, <release> and <releasenumber> represent the release version you are upgrading to, and <targetbuildnumber> represents the build number that you are upgrading to. The procedure includes optional steps to avoid losing any updates that are pushed to the /etc directory during the upgrade.

1. Use an SSH client, such as PuTTy, to open an SSH connection to the appliance.
2. Log on to the appliance by using the administrator credentials. Save the running configuration. At the prompt, type:

   ```
   save config
   ```

3. Switch to the shell prompt by running the following command:

   ```
   shell
   ```

4. Create a copy of the ns.conf file. At the shell prompt, type:

   a) `cd /nsconfig`
   b) `cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>`

   You should backup the configuration file to another computer.

5. (Optional) If you have modified some of the following files in the /etc directory, and copied them to /nsconfig to maintain persistency, any updates that are pushed to the /etc directory during the upgrade might be lost:

   - `ttys`
   - `resolv.conf`
   - `sshd\\_config`
   - `host.conf`
   - `newsyslog.conf`
   - `host.conf`
   - `httpd.conf`
   - `rc.conf`
   - `syslog.conf`
   - `crontab`
   - `monitrc`

   To avoid losing these updates, create a /var/nsconfig_backup directory, and move the customized files to this directory. That is, move any files that you modified in /etc directory and copied to /nsconfig by running the following command:

   ```
   cp /nsconfig/\<filename\> /var/nsconfig\\_backup
   ```

Example:

cp /nsconfig/syslog.conf /var/nsconfig_backup

6. Create a location for the installation package. At the shell prompt type:

- `cd /var/nsinstall`
- `mkdir \<releasenumber\>nsinstall`
- `cd \<releasenumber\>nsinstall`
- `mkdir build\\_\<targetbuildnumber\>`
- `cd build\\_\<targetbuildnumber\>`

1. Copy the already downloaded Citrix NetScaler firmware to the directory that you created for it in step 5, by using any file transferring method such as WinSCP. See the Before You Begin section for more information about downloading the Citrix NetScaler firmware.

2. Extract the contents of the installation package. Example:

```
tar -xvzf build-11.1-47.1_nc.tgz
```

1. Run the installns script to install the new version of the system software.

   ```
   ./installns
   ```

2. When prompted, restart the Citrix NetScaler.

3. (Optionally) If you've created a copy of the ns.conf file in the **Before You Begin** section, do the following:

   a) Manually compare the files in /var/nsconfig_backup and /etc and make appropriate changes in /etc.
   b) To maintain persistency, move the updated files in /etc to /nsconfig.
   c) Restart the appliance to put the changes into effect.

Following is an example of Citrix NetScaler firmware upgrade.

```
login: nsroot


Password: nsroot


Last login: Thu Jun 23 15:05:05 2016 from 10.252.243.134


Done


> save config
```

```


> shell


Last login:Thu Jun 23 15:05:05 2016 from 10.252.243.134

root@NSnnn# cd /var/nsinstall


root@NSnnn# cd 11.1nsinstall


root@NSnnn# mkdir build_47.10


root@NSnnn# cd build_47.10


root@NSnnn# ftp <FTP server IP address>


ftp> mget build-11.1-47.10_nc.tgz


ftp> bye


root@NSnnn# tar xzvf build-11.1-47.10_nc.tgz


root@NSnnn# ./installns


installns version (11.1-47.10) kernel (ns-11.1-47.10_nc.gz)


...


...


...

```

```

Copying ns-11.1-47.10_nc.gz to /flash/ns-11.1-47.10_nc.gz ...





...


Installation has completed.





Reboot NOW? [Y/N] Y
```

Upgrade a Citrix NetScaler standalone appliance by using NITRO API

To use NITRO API to upgrade or downgrade a Citrix NetScaler, see Automate Citrix NetScaler Upgrade and Downgrade with a Single API.

Directory locations of script files for user monitors

In release 10.1 build 122.17, the script files for user monitors are at a new location. If you upgrade an appliance or virtual appliance to release 10.1 build 122.17 or later, the changes are as follows:

- A new directory named conflicts is created in /nsconfig/monitors/ and all the built-in scripts of the previous builds are moved to this directory.
- All new built-in scripts are available in the /netscaler/monitors/ directory. All custom scripts are available in the /nsconfig/monitors/ directory.
- You must save a new custom script in the /nsconfig/monitors/ directory.
- After the upgrade is completed, if a custom script is created and saved in the /nsconfig/monitors/ directory with the same name as that of a built-in script, the script in the /netscaler/monitors/ directory takes priority. That is, the custom script is not run.

If you provision a virtual appliance running release 10.1 build 122.17 or later, the changes are as follows:

- All built-in scripts are available in the /netscaler/monitors/ directory
- The directory /nsconfig/monitors/ is empty.
- If you create a custom script, you must save it in the /nsconfig/monitors/ directory.

For more information about user monitors, see Understanding user monitors.

Check and install a Citrix NetScaler software update

Update the Citrix NetScaler software when an update is available, for a better performance. A Citrix NetScaler update can include feature improvements, performance fixes, or enhancements. Make sure you read the release notes to see what fixes and enhancements are available in the update. To check and install a software update, do the following.

1. In the Citrix NetScaler home page, click **Check for Update** from the **nsroot** menu at the top right corner of the page.
2. In the **Latest System Software Updates Available** page, check the available software update that you can install.
3. Click **Download** to download the installation package from the Citrix download website.
4. After you have downloaded the software package, install the update through either CLI or GUI procedure.

> **Note**
>
> The **Check for Update** link is accessible only if you log into the GUI through the HTTP protocol and not through the HTTPS protocol.

Downgrade a Citrix NetScaler standalone appliance

September 21, 2020

You can downgrade to any earlier release on a standalone Citrix NetScaler by using the CLI. Downgrading using the GUI is not supported.

> **Note**
>
> Loss in configuration might occur when downgrading. You should compare the configurations before and after the downgrade, and then manually reenter any missing entries.

Procedure to downgrade a standalone Citrix NetScaler

Follow the steps given below to downgrade a Citrix NetScaler standalone appliance running release 11.1 to an earlier release.

In this procedure, <release> and <releasenumber> represent the release version you are downgrading to, and <targetbuildnumber> represents the build number that you are downgrading to.

1. Open an SSH connection to the Citrix NetScaler by using an SSH client, such as PuTTY.

2. Log on to the Citrix NetScaler by using the administrator credentials. Save the running configuration. At the prompt, type:

   save config

3. Create a copy of the ns.conf file. At the shell prompt, type:

   a) `cd /nsconfig`
   b) `cp ns.conf ns.conf.NS<currentbuildnumber>`

   You should backup a copy of the configuration file on another computer.

4. Copy the <releasenumber> configuration file (ns.conf.NS<releasenumber>) to ns.conf. At the shell prompt, type:

   cp ns.conf.NS<releasenumber> ns.conf

   Note: ns.conf.NS<releasenumber> is the backup configuration file that is automatically created when the system software is upgraded from release version <releasenumber> to the current release version. There may be some loss in configuration when downgrading. After the appliance restarts, compare the configuration saved in step 3 with the running configuration, and make any adjustments for features and entities configured before the downgrade. Save the running configuration after making the changes.

   Important: If routing is enabled, perform step 5. Otherwise, skip to step 6.

5. If routing is enabled, the ZebOS.conf file will contain the configuration. At the shell prompt, type:

   a) cd /nsconfig
   b) cp ZebOS.conf ZebOS.conf.NS
   c) cp ZebOS.conf.NS<targetreleasenumber> ZebOS.conf

6. Change directory to /var/nsinstall/<releasenumber>nsinstall, or create one if it does not exist.

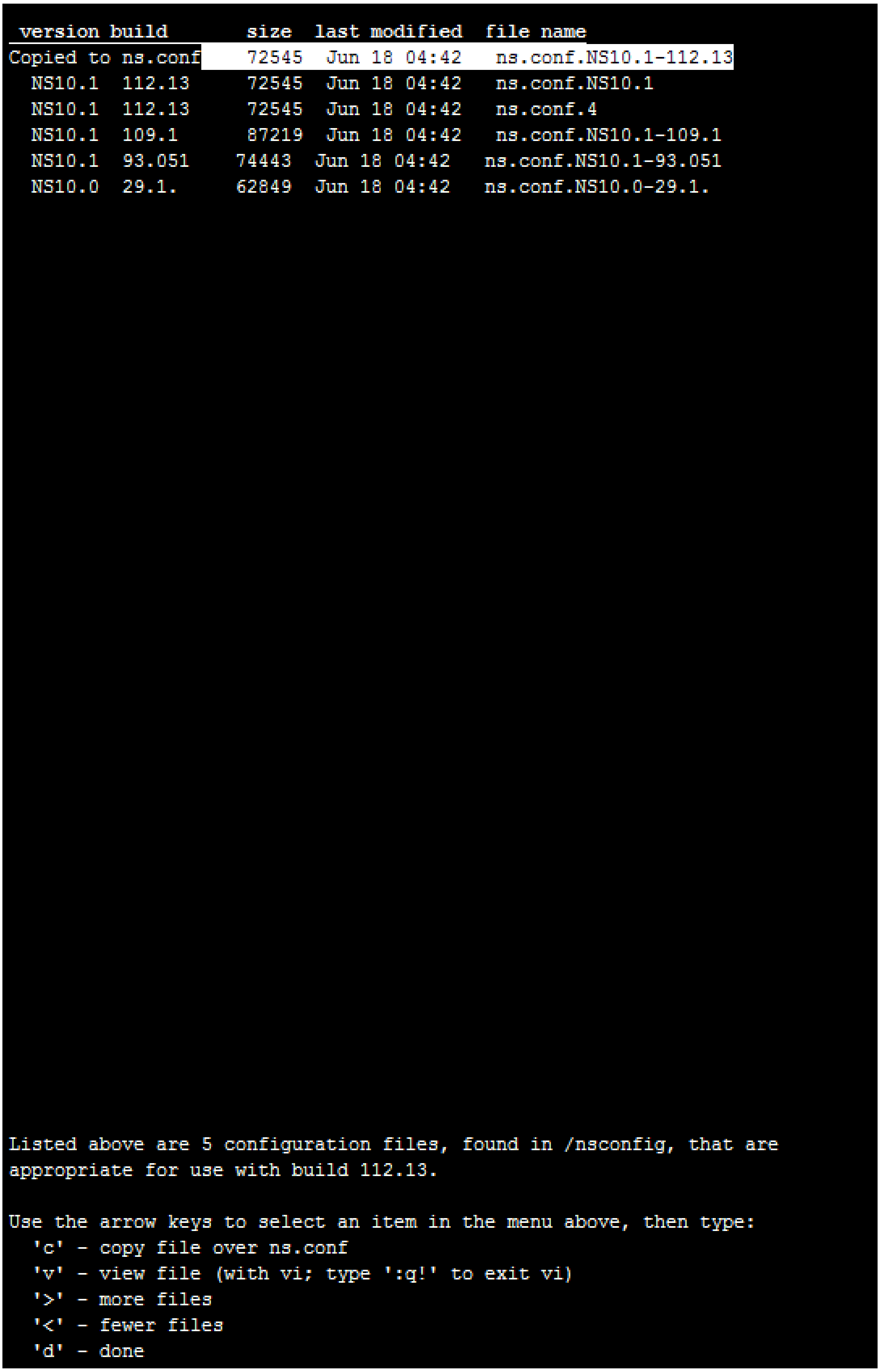
7. Change directory to build_<targetbuildnumber>, or create one if it does not exist.

8. Download or copy the installation package (build-<release>-<targetbuildnumber>.tgz) to this directory and extract the contents of the installation package.

9. Run the installns script to install the new version of the system software. The script updates the /etc directory.

   If the configuration file for the build that you are downgrading to exists on the appliance, you are prompted to load that configuration, as shown in the following figure.

   Figure 1. Downgrade menu if configuration file exists



```
 version build          size  last modified  file name
Copied to ns.conf        72545  Jun 18 04:42   ns.conf.NS10.1-112.13
  NS10.1  112.13         72545  Jun 18 04:42   ns.conf.NS10.1
  NS10.1  112.13         72545  Jun 18 04:42   ns.conf.4
  NS10.1  109.1          87219  Jun 18 04:42   ns.conf.NS10.1-109.1
  NS10.1  93.051        74443  Jun 18 04:42   ns.conf.NS10.1-93.051
  NS10.0  29.1.         62849  Jun 18 04:42   ns.conf.NS10.0-29.1.

Listed above are 5 configuration files, found in /nsconfig, that are
appropriate for use with build 112.13.

Use the arrow keys to select an item in the menu above, then type:
  'c' - copy file over ns.conf
  'v' - view file (with vi; type ':q!' to exit vi)
  '>' - more files
  '<' - fewer files
  'd' - done
```

If the free space available on the flash drive is insufficient to install the new build, the Citrix NetScaler aborts the installation. Manually clean up the flash drive and restart the installation.

Example:

```
login: nsroot

Password: nsroot

Last login: Mon Apr 24 02:06:52 2017 from 10.102.29.9

Done

> save config

> shell

root@NSnnn# cp ns.conf.NS10.5 ns.conf

root@NSnnn# cd /var/nsinstall

root@NSnnn# mkdir 10.5nsinstall

root@NSnnn# cd 10.5nsinstall

root@NSnnn# mkdir build_57

root@NSnnn# cd build_57

root@NSnnn# ftp 10.102.1.1

ftp> mget build-10.5-57_nc.tgz

ftp> bye

root@NSnnn# tar -xzvf build-10.1-125_nc.tgz

root@NSnnn# ./installns

installns version (10.5-57) kernel (ns-10.5-57.gz)

...

...

```

```
...

Copying ns-10.5-57.gz to /flash/ns-10.5-57_nc.gz ...

Changing /flash/boot/loader.conf for ns-10.5-57 ...



Installation has completed.



Reboot NOW? [Y/N] Y
```

Upgrade a high availability pair

December 10, 2020

One of the requirements of Citrix NetScaler appliances in a high availability setup is to install the same Citrix NetScaler software release on both appliances of the setup. Therefore, when software on one appliance is upgraded, ensure that the software is upgraded on both the appliances.

You can follow the same procedure to upgrade a standalone appliance or each appliance in a high availability pair, although additional considerations apply to upgrading a high availability pair.

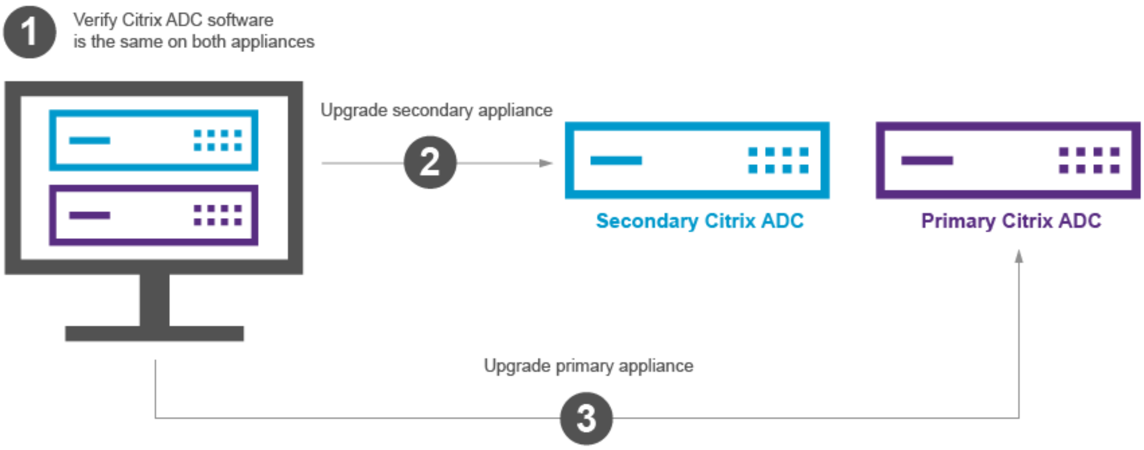
Before you start a Citrix NetScaler firmware upgrade on an HA pair, read the prerequisites mentioned in the Before you begin section. Also, you need to consider a few HA-specific points.

Points to note

- First upgrade the secondary node, and then the primary node. Upgrading software on the secondary appliance before the primary appliance ensures that the upgrade process is completed without any issues.
- If the two nodes in an HA configuration are running different Citrix NetScaler software releases, the following information does not get synchronized on the primary and secondary nodes:
    - Configuration propagation and synchronization
    - States of the services
    - Connection failover sessions
    - Persistence sessions

- The above information might not get synchronized on the primary and secondary nodes if the two nodes are running different builds of the same release. Refer to the Known Issues section of the release notes to check if your Citrix NetScaler build has this issue.
- Synchronization of the files in the All mode of the Sync HA files command works successfully if the two nodes in an HA configuration are running different Citrix NetScaler software releases, or the two nodes are running different builds of the same release. For more information, see Synchronising Configuration Files in High Availability Setup.

Figure. Upgrade a high availability pair


1
Verify Citrix ADC software is the same on both appliances
Upgrade secondary appliance
2
Secondary Citrix ADC
Primary Citrix ADC
Upgrade primary appliance
3

You can upgrade using the Citrix NetScaler CLI or GUI.

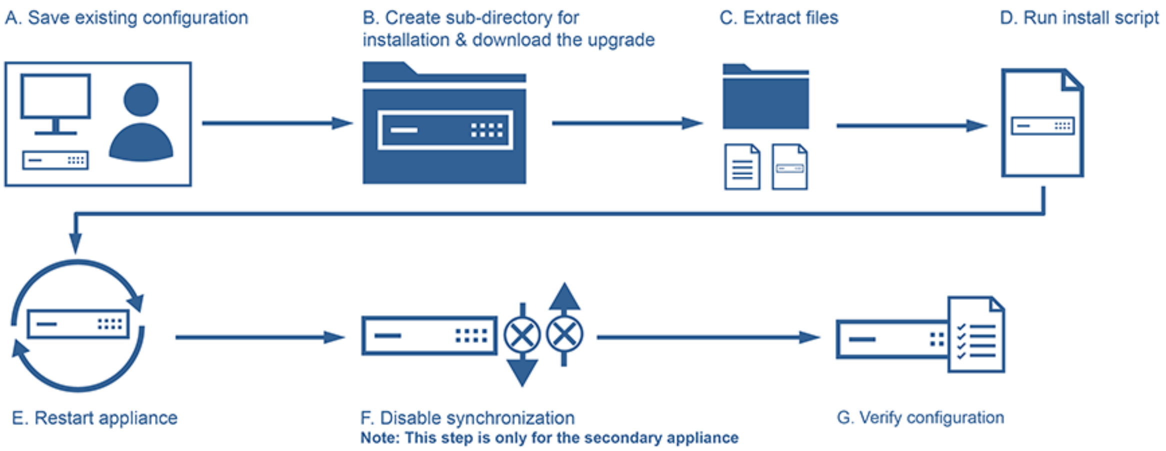
Upgrade a high availability pair by using the CLI

The upgrade process includes the following steps:

1. Upgrade software on the secondary appliance
2. Upgrade software on the upgrade appliance
3. Synchronize secondary appliance

Upgrade software on the secondary appliance

The following illustration depicts the procedure to upgrade software on the secondary appliance:


A. Save existing configuration
B. Create sub-directory for installation & download the upgrade
C. Extract files
D. Run install script
E. Restart appliance
F. Disable synchronization
Note: This step is only for the secondary appliance
G. Verify configuration

1. Log on to the secondary NetScaler appliance using an SSH utility, such as PuTTY and specifying the NetScaler IP (NSIP). Use the nsroot credentials to log on to the appliance.
2. From the command line interface of the appliance, type the following command to save the existing configuration: save config
3. Switch to the shell prompt.

```
login as: username
Using keyboard-interactive authentication.
Password:
Last login: Wed Jun 24 14:59:16 2015 from 10.252.252.65
Done
> shell
Copyright (c) 1992-20

<!--NeedCopy-->
```

4. Run the following command to change to the default installation directory: # cd /var/nsinstall
5. Run the following command to create a temporary subdirectory of the nsinstall directory: **# mkdir x_xnsinstall**

   > Note: The text x_x is used to name the NetScaler version for future configurations. For example, the directory for the installation files of NetScaler 9.3 us called 9_3nsinstall. Do not use a period (.) in the folder name, it can cause failed upgrades.

6. Change to the **x_xnsinstall** directory.
7. Download the required installation package and documentation bundle, such as “ns-x.0-xx.x-doc.tgz,” to the temporary directory created in Step 4.

> **Notes:**
>
> - Some builds do not have a documentation bundle as it does not have to be installed.
> - Click the **Documentation** tab from the GUI to access the documentation.

8. Before you run the install script, the files must be extracted and placed on the appliance. Use the following command to uncompress the bundle downloaded from Citrix website: **tar -zxvf ns-x.0-xx.x-doc.tgz**. The following is a quick explanation of the parameters used.

   x: Extract files

   v: Print the file names as they are extracted one by one

   z: The file is a "gzipped" file

   f: Use the following tar archive for the operation

9. Run the following command to install the downloaded software: # ./installns

   > Note: If the appliance does not have sufficient disk space to install the new kernel files, the installation process performs an automatic cleanup of the flash drive.

10. After the installation process is complete, the process prompts to restart the appliance. Press y to restart the appliance.

11. Log on to the appliance Command Line Interface using the nsroot credentials.

12. Run the following command from to display the state of the NetScaler appliance: **> show ha node** The output of the preceding command should indicate that the appliance is a secondary node and synchronization is disabled.

13. Run the following command to disable synchronization on the appliance if synchronization is not disabled: **> set ha node -hasync disabled**

14. Ensure that the configuration is complete and as expected.

15. Run the following command to perform a force failover and takeover as primary appliance: **> force failover**

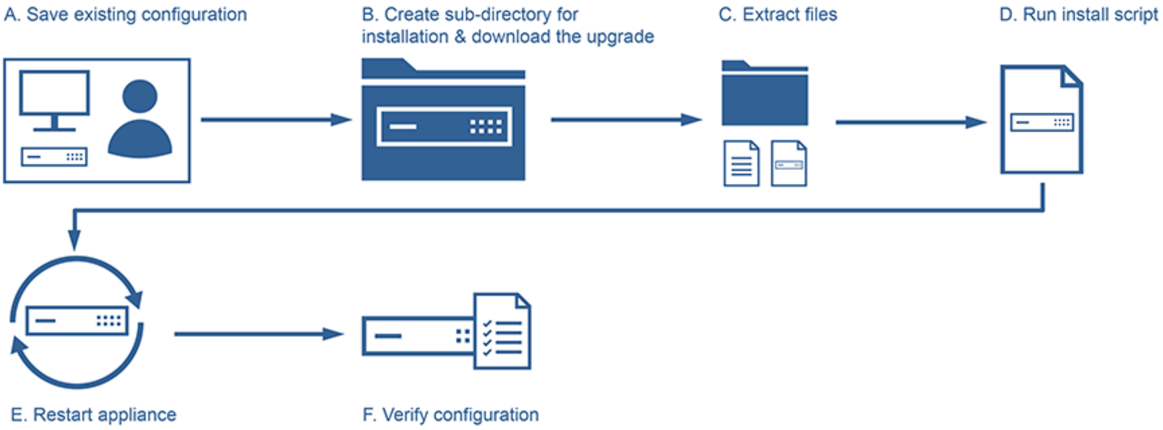
Here's a sample configuration in the new primary node.

```
login: nsroot
    Password: nsroot
    Last login: Monday Apr  17 08:37:26 2017 from 10.102.29.9
    Done
    show ha node
            2 nodes:
    1)      Node ID:      0
            IP:        10.0.4.2
            Node State: UP
```

```
                Master State: Primary
                ...
                Sync State: AUTO DISABLED
                Propagation: AUTO DISABLED
                ...
        Done
<!--NeedCopy-->
```

Upgrade software on the primary appliance

The following illustration depicts the procedure to upgrade software on the primary appliance:


A. Save existing configuration
B. Create sub-directory for installation & download the upgrade
C. Extract files
D. Run install script
E. Restart appliance
F. Verify configuration

> Note: After completing the "Upgrade software on the secondary appliance" procedure, the original primary appliance is now a secondary appliance.

1. Log on to the secondary NetScaler appliance using an SSH utility, such as PuTTY. Use the nsroot credentials to log on to the appliance. Follow the same steps as mentioned in the above section to complete the installation process. We have to follow the same steps as mentioned in step 2 to step 9 in the previous section(Upgrade software of the secondary appliance)
2. After the installation process is complete, the process prompts to restart the appliance. Press y to restart the appliance.
3. Log on to the appliance Command Line Interface using the nsroot credentials.
4. Run the following command to display the state of the NetScaler appliance: **> show ha node**. The output of the preceding command should indicate that the appliance is a secondary node and the status of the node state is marked as UP.
5. If the appliance is not a primary appliance, run the following command to perform a force failover to ensure that the appliance is a primary appliance: **> force failover**
6. Verify that the appliance is a primary appliance.

Enable synchronization on the secondary appliance

To enable synchronization on the secondary appliance, complete the following procedure:

1. Run the following command to verify that the appliance is a secondary appliance: **> show node**
2. Run the following command to enable synchronization on the appliance: **> set ha node -hasync enabled**
3. Run the following command to verify that the configuration of the secondary appliance is synchronized with that of the primary appliance: **> show ns runningconfig**

Here's an example configuration of the new primary node and the new secondary node.

```
show ha node
        Node ID:      0
        IP:   10.0.4.2
        Node State: UP
        Master State: Primary
...
...
        INC State: DISABLED
        Sync State: ENABLED
        Propagation: ENABLED
        Enabled Interfaces : 1/1
        Disabled Interfaces : None
        HA MON ON Interfaces : 1/1
...
...
        Local node information
        Critical Interfaces: 1/1
Done

Show ha node
        Node ID:      0
        IP:   10.0.4.11
        Node State: UP
        Master State: Secondary
        ..
      ..
        INC State: DISABLED
        Sync State: SUCCESS
        Propagation: ENABLED
        Enabled Interfaces : 1/1
        Disabled Interfaces : None
        HA MON ON Interfaces : 1/1
        . . .
```

```
            . . .
            Local node information:
            Critical Interfaces: 1/1
    Done
<!--NeedCopy-->
```

Upgrade a high availability pair by using the GUI

Follow these steps to upgrade a Citrix NetScaler pair in a high availability setup, by using the GUI:

1. Log on to the secondary node and perform the upgrade as described in the section "Upgrade a Citrix NetScaler standalone appliance by using the GUI" in Upgrade a Citrix NetScaler standalone appliance.
2. Log on to the primary node and perform the upgrade as described in the section "Upgrade a Citrix NetScaler standalone appliance by using the GUI" in Upgrade a Citrix NetScaler standalone appliance.

Downgrade a high availability pair

September 21, 2020

You can downgrade to any release on a high availability pair by using the command line interface. The GUI does not support the downgrade process.

To downgrade the system software on a Citrix NetScaler pair in a high availability pair, you need to downgrade the software first on the secondary node and then on the primary node. For instructions on downgrading each node separately, see Downgrade a Citrix NetScaler standalone appliance.

> **Important**
>
> Loss in configuration might occur when downgrading. You should compare the configurations before and after the downgrade, and then manually re-enter any missing entries.

Troubleshooting

September 21, 2020

If the appliance does not work as expected after you complete the installation, upgrade, or downgrade process, the first thing to do is to check for the most common causes of the problem.

Resources for troubleshooting

For best results, use the following resources to troubleshoot an issue related to installing, upgrading, or downgrading a Citrix NetScaler:

- The configuration files from the appliance. In case of a High Availability pair, the configuration files from both appliances.
- The following files from the appliance(s):
  - The relevant newnslog files.
  - The ns.log file.
  - The messages file.
- A network topology diagram.

Troubleshooting issues related to the installation, upgrade, and downgrade processes

Following are the most common installation, upgrade, and downgrade issues, and tips for resolving them:

- **Issue**

  The Citrix NetScaler is not accessible after the software downgrade

  Cause

  During the software downgrade process, if the configuration file of the existing release and build does not match the configuration file of the earlier release and build, the appliance cannot load the configuration, and the default IP address is assigned to the appliance.

  Resolution

  - Verify that the appliance is accessible from the console.
  - Verify the NSIP address and the routes on the appliance.
    * If the IP address has changed to the default 192.168.100.1 IP address, change the IP address as required.
    * Verify that the appliance is accessible.
- **Issue**

  Configuration of the appliance is lost after you upgrade the software across multiple releases.

  Cause

  Some migration commands are built in for upgrading to the next release. Such commands might not be available across releases.

  Resolution

  - Verify the path of the upgrade process. Citrix recommends upgrading by one release at a time. For example, if the software needs to be upgraded from Citrix NetScaler (previously

NetScaler) release 9.2 to release 10.1, the following is the recommended path for the upgrade:
    - Release 9.2 to release 9.3
    - Release 9.3 to release 10
    - Release 10 to release 10.1
  - Verify that the appliance has appropriate license files.
  - Verifying the configuration at each step of the upgrade process can give you pointers to the issue.

- **Issue**

  During an upgrade, if I run the command for synchronizing, the following message appears:

  Command failed on the secondary node but succeeded on the primary node.

  Resolution

  Do not run any dependent commands (set /unset /bind /unbind) when High Availability (HA) synchronization is in progress.

- **Issue**

  During an upgrade process, traffic does not pass through the new primary node when you run the force failover command.

  Resolution

  - Check for problems with the network topology and the switch configurations.
  - Run the set L2param -garpreply ENABLED command to enable the GARP reply.
  - Try using virtual MAC if not already used.
  - Run the sendarp –a command from the primary node.

- **Issue**

  In an HA pair, after you run the force HA failover command the devices keep rebooting. The secondary device does not come up after an upgrade.

  Resolution

  Check to see if the /var directory is full. If so, remove the old installation files. Run the df –h command to show the available disk space.

- **Issue**

  After upgrading an HA pair, one of the nodes is listed as state UNKNOWN.

  Resolution

  - Check to see if both nodes are running the same build. If the builds are not same and HA nodes have a version mismatch, some of the fields are shown as UNKNOWN when you run the show ha node command.

    - Check to see if the secondary appliance is reachable.

- **Issue**

  After you upgrade the Citrix NetScaler, the interface shows most of the load balancing virtual servers and services are DOWN.

  Resolution

  Verify that the SNIP address is active on the secondary appliance. Also, type the show service command to see if the service is running.

- **Issue**

  After performing an upgrade, all virtual servers are down on the secondary appliance.

  Resolution

  Enable the HA state and HA synchronization by running the following commands:

    - set node hastate enable
    - set node hasync enable

  Disabling HA is not recommended.

- **Issue**

  After performing a downgrade, the Citrix NetScaler does not boot up properly.

  Resolution

  Check to see if the correct license has been installed.

- **Issue**

  In an HA pair, some features do not get synchronized after an upgrade is performed.

  Resolution

  Run the sync ha file misc command to synchronize the configurations files from the primary node to secondary node.

- **Issue**

  During reboot, the following error message appears:

  One or more commands in ns.conf failedWhat should I do?

  Resolution

  Make sure that no command in the ns.conf file exceeds the 255 byte limit. In commands that create policies that are too long for the 255-byte limit, you can use pattern sets to shorten the policies.

  Example:

“‘ pre codeblock
add cs policy p11 -rule ‘HTTP.REQ.URL.ENDSWITH_ANY(“ctx_file_extensions”)’
Done

~~~
ctx\_file\_extensions is a default patset that covers a large number of extensions. In addition to the default pattern sets, you can create user-defined pattern sets. Add a patset by running the following command:

``` pre codeblock
add patset <name>
<!--NeedCopy-->
~~~

Note: Patsets are supported only in release 9.3 or later.

- **Issue**

  When upgrading a Citrix NetScaler VPX appliance, I am told to free up space in /var. What files do I remove?

  Resolution

  Remove the old installation files from /var/tmp/ directory. Also remove unwanted files from /flash.

- **Issue**

  There is no connectivity to the graphical user interface (GUI) when you run the force HA failover command on the secondary appliance.

  Resolution

  Log on to the secondary appliance using the command line interface and enable the access to GUI by running the set ns ip <IP> -gui enabled command.

- **Issue**

  After performing an upgrade, and when I click on any link on the GUI that has to load a java applet (Upgrade Wizard or license Wizard), the following error message appears: **GUI version does not match with the kernel version. Please close this instance, clear java plug-in cache and reopen**.

  Resolution

  - Log on to the Citrix NetScaler using the GUI.
  - Navigate to Citrix NetScaler Gateway > Global Settings.
  - Click Change Global Settings under Settings.
  - In the details pane, under Client Experience, select Default from the UI theme list.
  - Click OK.

> **Note**
>
> These troubleshooting steps also apply to issues with configuration loss when downgrading the software across multiple releases.
>
> For any other issue, see the release notes, Knowledge Center articles, and FAQs.

FAQs

September 21, 2020

For answers to the questions that you might have about upgrading the Citrix NetScaler firmware, see the Installing, Upgrading, and Downgrading FAQs.

New and Deprecated Commands, Parameters, and SNMP OIDs

September 21, 2020

Jun 17, 2016

This section lists the new and deprecated commands, parameters, and SNMP OIDS.

New Commands

The following table lists the new commands in release 11.1.

Command Group	Command
Appflow	set appflow collector
Network	set vrID6, set netbridge, bind netProfile, unbind netProfile
NS	flush ns sourceroutecachetable
Protocol	clear protocol httpBand
Router	config router dynamicRouting
SSL	stat ssl vserver
Reputation	set reputation settings, show reputation settings

New Parameters

The following table lists the new parameters in release 11.1.

Command group: Basic

Commands:

- `rm server \[-removeGSLBSvcOnly\]`
- `add service \[-monConnectionClose\]`
- `show service \[-monConnectionClose\]`
- `add serviceGroup \[-monConnectionClose\]`
- `show serviceGroup \[-monConnectionClose\]`
- `start nstrace \[-skipLocalSSH\]`
- `start nstrace \[-capsslkeys\]`
- `show nstrace \[-skipLocalSSH\]`
- `show nstrace \[-capsslkeys\]`

Command group: AAA

Commands:

- `show aaa session \[-aaaUserConnInfo\]`
- `set aaa radiusParams \[-authservRetry\]`
- `show aaa radiusParams \[-authservRetry\]`
- `set aaa parameter \[-aaaSessionLoglevel\]`
- `show aaa parameter \[-aaaSessionLoglevel\]`

Command group: AppFlow

Commands:

- `add appflow action \[-pageTracking\]`
- `add appflow action \[-securityInsight\]`
- `set appflow action \[-pageTracking\]`
- `set appflow action \[-securityInsight\]`
- `show appflow action \[-pageTracking\]`
- `show appflow action \[-securityInsight\]`
- `set appflow param \[-SecurityInsightRecordInterval\]`

- `set appflow param \[-subscriberAwareness\]`
- `set appflow param \[-SecurityInsightTraffic\]`
- `set appflow param \[-cacheInsight\]`
- `show appflow param \[-SecurityInsightRecordInterval\]`
- `show appflow param \[-SecurityInsightTraffic\]`
- `show appflow param \[-subscriberAwareness\]`
- `show appflow param \[-cacheInsight\]`

Command group: Audit

Commands:

- `add audit syslogAction \[-serverDomainName\]`
- `set audit syslogAction \[-serverDomainName\]`
- `set audit syslogAction \[-domainResolveRetry\]`
- `set audit syslogAction \[-domainResolveNow\]`
- `show audit syslogAction \[-serverDomainName\]`
- `show audit syslogAction \[-IP\]`
- `show audit syslogAction \[-domainResolveRetry\]`
- `add audit nslogAction \[-serverDomainName\]`
- `set audit nslogAction \[-serverDomainName\]`
- `set audit nslogAction \[-domainResolveRetry\]`
- `set audit nslogAction \[-domainResolveNow\]`
- `show audit nslogAction \[-serverDomainName\]`
- `show audit nslogAction \[-domainResolveRetry\]`
- `show audit nslogAction \[-IP\]`

Command group: Authentication

Commands:

- `add authentication radiusAction \[-authservRetry\]`
- `set authentication radiusAction \[-authservRetry\]`
- `show authentication radiusAction \[-authservRetry\]`
- `add authentication negotiateAction \[-NTLMPath\]`

- `set authentication negotiateAction \[-NTLMPath\]`
- `show authentication negotiateAction \[-NTLMPath\]`
- `add authentication samlAction \[-skewTime\]`
- `set authentication samlAction \[-skewTime\]`
- `show authentication samlAction \[-skewTime\]`
- `bind authentication vserver \[-portaltheme\]`
- `unbind authentication vserver \[-portaltheme\]`
- `show authentication vserver \[-portaltheme\]`
- `add authentication samlIdPProfile \[-samlBinding\]`
- `add authentication samlIdPProfile \[-skewTime\]`
- `add authentication samlIdPProfile \[-signAssertion\]`
- `add authentication samlIdPProfile \[-keyTransportAlg\]`
- `set authentication samlIdPProfile \[-samlBinding\]`
- `set authentication samlIdPProfile \[-skewTime\]`
- `set authentication samlIdPProfile \[-signAssertion\]`
- `set authentication samlIdPProfile \[-keyTransportAlg\]`
- `show authentication samlIdPProfile \[-samlBinding\]`
- `show authentication samlIdPProfile \[-skewTime\]`
- `show authentication samlIdPProfile \[-signAssertion\]`
- `show authentication samlIdPProfile \[-keyTransportAlg\]`
- `add authentication loginSchema \[-builtin\]`
- `show authentication loginSchema \[-builtin\]`

Command group: Cluster

Commands:

- `show cluster instance \[-validMtu\]`
- `show cluster node \[-routeMonitor\]`
- `show cluster node \[-routeMonState\]`
- `show cluster node \[-netmask\]`
- `bind cluster node \[-routeMonitor\]`

- `unbind cluster node \[-routeMonitor\]`

Command group: DNS

Command:

- `show dns naptrRec \[-vServerName\]`

Command group: GSLB

Commands:

- `add gslb site \[-naptrReplacementSuffix\]`
- `set gslb site \[-naptrReplacementSuffix\]`
- `show gslb site \[-naptrReplacementSuffix\]`
- `add gslb service \[-naptrReplacement\]`
- `add gslb service \[-naptrOrder\]`
- `add gslb service \[-naptrServices\]`
- `add gslb service \[-naptrDomainTTL\]`
- `add gslb service \[-naptrPreference\]`
- `set gslb service \[-naptrOrder\]`
- `set gslb service \[-naptrPreference\]`
- `set gslb service \[-naptrServices\]`
- `set gslb service \[-naptrReplacement\]`
- `show gslb service \[-lastresponse\]`
- `show gslb service \[-naptrOrder\]`
- `show gslb service \[-naptrPreference\]`
- `show gslb service \[-naptrServices\]`
- `show gslb service \[-naptrReplacement\]`
- `show gslb service \[-naptrDomainTTL\]`
- `add gslb vserver \[-ECS\]`
- `add gslb vserver \[-ecsAddrValidation\]`
- `set gslb vserver \[-ECS\]`
- `set gslb vserver \[-ecsAddrValidation\]`
- `show gslb vserver \[-ECS\]`

- `show gslb vserver \[-ecsAddrValidation\]`

Command group: HA

Command:

- `show HA node \[-haHeartbeatifaces\]`

Command group: ICA
Commands:

- `add ica action \[-latencyprofileName\]`
- `set ica action \[-latencyprofileName\]`
- `show ica action \[-latencyprofileName\]`

Command group: IPsec
Commands:

- `add ipsec profile \[-responderOnly\]`
- `show ipsec profile \[-responderOnly\]`
- `set ipsec parameter \[-responderOnly\]`
- `unset ipsec parameter \[-responderOnly\]`
- `show ipsec parameter \[-responderOnly\]`

Command group: Load balancing

Commands:

- `add lb monitor \[-builtin\]`
- `add lb monitor \[-sslProfile\]`
- `set lb monitor \[-sslProfile\]`
- `unset lb monitor \[-sslProfile\]`
- `bind lb monitor \[-certkeyName\]`
- `unbind lb monitor \[-certkeyName\]`
- `show lb monitor \[-builtin\]`
- `show lb monitor \[-sslProfile\]`
- `show lb monitor \[-certkeyName\]`
- `show lb monitor \[-CA\]`
- `show lb monitor \[-crlCheck\]`
- `show lb monitor \[-ocspCheck\]`

- `add lb vserver \[-lbprofilename\]`
- `add lb vserver \[-redirectFromPort\]`
- `add lb vserver \[-httpsRedirectUrl\]`
- `set lb vserver \[-lbprofilename\]`
- `set lb vserver \[-redirectFromPort\]`
- `set lb vserver \[-httpsRedirectUrl\]`
- `unset lb vserver \[-lbprofilename\]`
- `unset lb vserver \[-redirectFromPort\]`
- `unset lb vserver \[-httpsRedirectUrl\]`
- `show lb vserver \[-backupLBMethod\]`
- `show lb vserver \[-preferredLocation\]`
- `show lb vserver \[-lbprofilename\]`
- `show lb vserver \[-redirectFromPort\]`
- `show lb vserver \[-httpsRedirectUrl\]`
- `set lb parameter \[-AllowBoundSvcRemoval\]`
- `set lb parameter \[-retainservicestate\]`
- `show lb parameter \[-AllowBoundSvcRemoval\]`
- `show lb parameter \[-retainservicestate\]`

Command group: LSN

Commands:

- `bind lsn group \[-pcpServer\]`
- `bind lsn group \[-logProfileName\]`
- `unbind lsn group \[-pcpServer\]`
- `unbind lsn group \[-logProfileName\]`
- `show lsn group \[-pcpServer\]`
- `show lsn group \[-logProfileName\]`
- `add lsn logprofile \[-logCompact\]`
- `set lsn logprofile \[-logCompact\]`
- `show lsn logprofile \[-logCompact\]`

- `set lsn parameter \[-subscrSessionRemoval\]`
- `show lsn parameter \[-subscrSessionRemoval\]`
- `show lsn session \[-IPv6Address\]`
- `show lsn deterministicNat \[-network6\]`
- `show lsn deterministicNat \[-natprefix\]`
- `show lsn deterministicNat \[-nattype\]`

Command group: Network
Commands:

- `add channel \[-haHeartbeat\]`
- `set channel \[-haHeartbeat\]`
- `show channel \[-haHeartbeat\]`
- `add vrID \[-preemptiondelaytimer\]`
- `show vrID \[-preemptiondelaytimer\]`
- `add vrID6 \[-priority\]`
- `add vrID6 \[-preemption\]`
- `add vrID6 \[-sharing\]`
- `add vrID6 \[-preemptiondelaytimer\]`
- `add vrID6 \[-trackifNumPriority\]`
- `bind vrID6 \[-trackifNum\]`
- `unbind vrID6 \[-trackifNum\]`
- `show vrID6 \[-preemption\]`
- `show vrID6 \[-sharing\]`
- `show vrID6 \[-preemptiondelaytimer\]`
- `show vrID6 \[-trackifNum\]`
- `show vrID6 \[-trackifNumPriority\]`
- `show vrID6 \[-effectivePriority\]`
- `add inat \[-tcpproxy\]`
- `add inat \[-ftp\]`
- `add inat \[-tftp\]`
- `add inat \[-usip\]`

- `add inat \[-usnip\]`
- `add inat \[-proxyIP\]`
- `set inat \[-tcpproxy\]`
- `set inat \[-ftp\]`
- `set inat \[-tftp\]`
- `set inat \[-usip\]`
- `set inat \[-usnip\]`
- `set inat \[-proxyIP\]`
- `show inat \[-proxyIP\]`
- `show inat \[-tcpproxy\]`
- `show inat \[-ftp\]`
- `show inat \[-tftp\]`
- `show inat \[-usip\]`
- `show inat \[-usnip\]`
- `show bridgegroup \[-ownerGroup\]`
- `add netbridge \[-vxlanVlanMap\]`

show netbridge [-vxlanVlanMap]

show netProfile [-srcPortRange]

show netProfile [-stateflag]

show netProfile [-flags]

show netProfile [-natRule]

show netProfile [-netmask]

set interface [-haHeartbeat]

set interface [-trunkmode]

set interface [-trunkAllowedVlan]

show interface [-haHeartbeat]

show interface [-trunkmode]

show interface [-trunkAllowedVlan]

set rnat [-connfailover]

unset rnat [-connfailover]

show rnat [-connfailover]

set L2Param [-stopMacMoveUpdate]

show L2Param [-stopMacMoveUpdate]

set L3Param [-ipv6DynamicRouting]

show L3Param [-ipv6DynamicRouting]

add forwardingSession [-sourceroutecache]

set forwardingSession [-sourceroutecache]

show forwardingSession [-sourceroutecache]

add vxlan [-type is introduced]

add vxlan [-innerVlanTagging]

set vxlan [-innerVlanTagging]

show vxlan [-innerVlanTagging]

show vxlan [-type]

show vxlan [-protocol]

Command group: NS

Commands:

- `show ns acl \[-aclassociate\]`
- `show ns acl6 \[-aclassociate\]`
- `add ns ip6 \[-vrID6\]`
- `set ns ip6 \[-vrID6\]`
- `show ns ip6 \[-vrID6\]`
- `add ns tcpProfile \[-tcpFastOpen\]`
- `add ns tcpProfile \[-Hystart\]`
- `add ns tcpProfile \[-dupackthresh\]`
- `set ns tcpProfile \[-tcpFastOpen\]`
- `set ns tcpProfile \[-Hystart\]`
- `set ns tcpProfile \[-dupackthresh\]`
- `unset ns tcpProfile \[-tcpFastOpen\]`

- `unset ns tcpProfile` \[-Hystart\]
- `unset ns tcpProfile` \[-dupackthresh\]
- `show ns tcpProfile` \[-tcpFastOpen\]
- `show ns tcpProfile` \[-Hystart\]
- `show ns tcpProfile` \[-dupackthresh\]
- `show ns license` \[-isSGwyLic\]
- `show ns license` \[-Rep\]
- `clear ns config` \[-RBAconfig\]
- `show ns connectiontable` \[-maxRcvbuf\]
- `show ns connectiontable` \[-linkmaxRcvbuf\]
- `show ns connectiontable` \[-RxQsize\]
- `show ns connectiontable` \[-linkRxQsize\]
- `show ns connectiontable` \[-maxSndbuf\]
- `show ns connectiontable` \[-linkmaxSndbuf\]
- `show ns connectiontable` \[-TxQsize\]
- `show ns connectiontable` \[-linkTxQsize\]
- `show ns connectiontable` \[-flavor\]
- `show ns connectiontable` \[-linkflavor\]
- `show ns connectiontable` \[-bwEstimate\]
- `show ns connectiontable` \[-linkbwEstimate\]
- `show ns connectiontable` \[-rttMin\]
- `show ns connectiontable` \[-linkrttMin\]
- `show ns connectiontable` \[-name\]
- `show ns connectiontable` \[-linkName\]
- `show ns connectiontable` \[-tcpmode\]
- `show ns connectiontable` \[-linktcpmode\]
- `show ns connectiontable` \[-realTimeRtt\]
- `show ns connectiontable` \[-linkrealTimeRtt\]
- `show ns connectiontable` \[-sndBuf\]

- `show ns connectiontable \[-linksndBuf\]`
- `show ns connectiontable \[-nsbTcpwaitQ\]`
- `show ns connectiontable \[-linknsbTcpwaitQ\]`
- `show ns connectiontable \[-nsbRetxQ\]`
- `show ns connectiontable \[-linknsbRetxQ\]`
- `show ns connectiontable \[-sackblocks\]`
- `show ns connectiontable \[-linksackblocks\]`
- `show ns connectiontable \[-congstate\]`
- `show ns connectiontable \[-linkcongstate\]`
- `show ns connectiontable \[-sndrecoverle\]`
- `show ns connectiontable \[-linksndrecoverle\]`
- `show ns feature \[-Rep\]`
- `show ns mode \[-ULFD\]`
- `set ns tcpParam \[-tcpFastOpenCookieTimeout\]`
- `show ns tcpParam \[-tcpFastOpenCookieTimeout\]`
- `unset ns pbr6 \[-vxlanVlanMap\]`
- `show ns pbr6 \[-vxlanVlanMap\]`
- `add ns partition \[-pmacInternal\]`
- `show ns partition \[-pmacInternal\]`

Command group: Protocol
Commands:

- `show protocol httpBand \[-reqBandSize\]`
- `show protocol httpBand \[-respBandSize\]`

Command group: RDP
Commands:

- `add rdp clientprofile \[-redirectComPorts\]`
- `add rdp clientprofile \[-redirectPnpDevices\]`
- `add rdp clientprofile \[-rdpListener\]`
- `set rdp clientprofile \[-redirectComPorts\]`
- `set rdp clientprofile \[-redirectPnpDevices\]`

- `set rdp clientprofile \[-rdpListener\]`
- `show rdp clientprofile \[-redirectComPorts\]`
- `show rdp clientprofile \[-redirectPnpDevices\]`
- `show rdp clientprofile \[-rdpListener\]`

Command group: Rewrite
Commands:

- `set rewrite param \[-timeout\]`
- `unset rewrite param \[-timeout\]`
- `show rewrite param \[-timeout\]`

Command group: SNMP
Commands:

- `set snmp option \[-snmpTrapLoggingLevel\]`
- `show snmp option \[-snmpTrapLoggingLevel\]`

Command group: SSL
Commands:

- `show ssl certKey \[-CertificateType\]`
- `create ssl certReq \[-digestMethod\]`
- `show ssl profile \[-sslpfobjecttype\]`

Command group: System

Commands:

- `add system user \[-maxsession\]`
- `set system user \[-maxsession\]`
- `show system user \[-maxsession\]`
- `set system parameter \[-totalAuthTimeout\]`
- `set system parameter \[-cliLoglevel\]`
- `show system parameter \[-totalAuthTimeout\]`
- `show system parameter \[-cliLoglevel\]`
- `restore system backup \[-skipbackup\]`
- `show system file \[-filesize\]`

Command group: Subscriber
Commands:

- `set subscriber param \[-idleTTL\]`
- `set subscriber param \[-idleAction\]`
- `set subscriber param \[-ipv6PrefixLookupList\]`
- `set subscriber param \[-builtin\]`
- `show subscriber param \[-idleTTL\]`
- `show subscriber param \[-idleAction\]`
- `show subscriber param \[-ipv6PrefixLookupList\]`
- `show subscriber param \[-builtin\]`
- `set subscriber gxInterface \[-revalidationTimeout\]`
- `show subscriber gxInterface \[-revalidationTimeout\]`

Command group: TM
Commands:

- `add tm sessionPolicy \[-expressionType\]`
- `show tm sessionPolicy \[-stateflag\]`
- `show tm sessionPolicy \[-expressionType\]`
- `show tm sessionPolicy \[-hits\]`
- `show tm sessionPolicy \[-gotoPriorityExpression\]`
- `add tm samlSSOProfile \[-skewTime\]`
- `set tm samlSSOProfile \[-skewTime\]`
- `show tm samlSSOProfile \[-skewTime\]`
- `show tm global \[-gotoPriorityExpression\]`

Command group: Utility

Commands:

- `show callhome \[-proxyAuthService\]`
- `set callhome \[-proxyAuthService\]`

Command group: VPN

Commands:

- `add vpn vserver \[-authnProfile\]`
- `add vpn vserver \[-vserverFqdn\]`
- `set vpn vserver \[-authnProfile\]`

- `set vpn vserver \[-vserverFqdn\]`
- `show vpn vserver \[-authnProfile\]`
- `show vpn vserver \[-vserverFqdn\]`
- `add vpn samlSSOProfile \[-skewTime\]`
- `set vpn samlSSOProfile \[-skewTime\]`
- `show vpn samlSSOProfile \[-skewTime\]`
- `add vpn sessionAction \[-sfGatewayAuthType\]`
- `add vpn sessionAction \[-alwaysONProfileName\]`
- `set vpn sessionAction \[-sfGatewayAuthType\]`
- `set vpn sessionAction \[-alwaysONProfileName\]`
- `show vpn sessionAction \[-sfGatewayAuthType\]`
- `show vpn sessionAction \[-alwaysONProfileName\]`
- `set vpn parameter \[-alwaysONProfileName\]`
- `unset vpn parameter \[-alwaysONProfileName\]`
- `show vpn parameter \[-alwaysONProfileName\]`
- `add vpn alwaysONProfile \[-internetAccess\]`
- `add vpn alwaysONProfile \[-clientControl\]`
- `add vpn alwaysONProfile \[-locationBasedVPN\]`
- `set vpn alwaysONProfile \[-internetAccess\]`
- `set vpn alwaysONProfile \[-clientControl\]`
- `set vpn alwaysONProfile \[-locationBasedVPN\]`
- `show vpn alwaysONProfile \[-internetAccess\]`
- `show vpn alwaysONProfile \[-clientControl\]`
- `show vpn alwaysONProfile \[-locationBasedVPN\]`

Command group: Application firewall

Command:

- `show appfw fieldType \[-all\]`

Deprecated Commands

The following table lists the commands that are deprecated in version 11.1.

Command group: Network

Commands:

- `add vpath`
- `rm vpath`
- `show vpath`
- `stat vpath`
- `set vPathParam`
- `show vPathParam`

Deprecated Parameters

The following table lists the parameters that are deprecated in version 11.1.

Command group: VPB

Commands:

- `add vpn vserver \[-userDomains\]`
- `set vpn vserver \[-userDomains\]`
- `show vpn vserver \[-userDomains\]`
- `add vpn sessionAction \[-clientOptions\]`
- `add vpn sessionAction \[-clientDebug\]`
- `set vpn sessionAction \[-clientOptions\]`
- `set vpn sessionAction \[-clientDebug\]`
- `show vpn sessionAction \[-clientOptions\]`
- `show vpn sessionAction \[-clientDebug\]`
- `set vpn parameter \[-clientOptions\]`
- `set vpn parameter \[-clientDebug\]`
- `set vpn parameter \[-userDomains\]`
- `unset vpn parameter \[-clientOptions\]`
- `unset vpn parameter \[-userDomains\]`

- `show vpn parameter \[-clientOptions\]`
- `show vpn parameter \[-clientDebug\]`
- `show vpn parameter \[-userDomains\]`

NEW SNMP OIDs

- pbrTotNullDrop,1.3.6.1.4.1.5951.4.1.1.22.5.25
- pbr6TotNullDrop,1.3.6.1.4.1.5951.4.1.1.22.7.25
- tcpOptimizationEnabled,1.3.6.1.4.1.5951.4.1.1.46.131
- tcpOptimizationBypassed,1.3.6.1.4.1.5951.4.1.1.46.132
- sslTotECDSAAuthorizations,1.3.6.1.4.1.5951.4.1.1.47.366
- nsLsnNAT64GlobalStatsGroup,1.3.6.1.4.1.5951.4.1.1.83.6
- lsnTotNAT64RxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.1
- lsnTotNAT64RxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.2
- lsnTotNAT64TxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.3
- lsnTotNAT64TxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.4
- lsnCurNAT64sessions,1.3.6.1.4.1.5951.4.1.1.83.6.5
- lsnNAT64CurSubscribers,1.3.6.1.4.1.5951.4.1.1.83.6.6
- lsnTotNAT64TcpRxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.7
- lsnTotNAT64TcpTxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.8
- lsnTotNAT64UdpRxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.9
- lsnTotNAT64UdpTxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.10
- lsnTotNAT64IcmpRxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.11
- lsnTotNAT64IcmpTxPkts,1.3.6.1.4.1.5951.4.1.1.83.6.12
- lsnTotNAT64TcpRxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.13
- lsnTotNAT64TcpTxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.14
- lsnTotNAT64UdpRxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.15
- lsnTotNAT64UdpTxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.16
- lsnTotNAT64IcmpRxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.17
- lsnTotNAT64IcmpTxBytes,1.3.6.1.4.1.5951.4.1.1.83.6.18
- lsnTotNAT64TcpDrpPkts,1.3.6.1.4.1.5951.4.1.1.83.6.19
- lsnTotNAT64UdpDrpPkts,1.3.6.1.4.1.5951.4.1.1.83.6.20
- lsnTotnat64IcmpDrpPkts,1.3.6.1.4.1.5951.4.1.1.83.6.21
- lsnCurNAT64TcpSessions,1.3.6.1.4.1.5951.4.1.1.83.6.22
- lsnCurNAT64UdpSessions,1.3.6.1.4.1.5951.4.1.1.83.6.23
- lsnCurNAT64IcmpSessions,1.3.6.1.4.1.5951.4.1.1.83.6.24
- lsnNAT64SessionsRate,1.3.6.1.4.1.5951.4.1.1.83.6.25
- lsnNAT64TcpSessionsRate,1.3.6.1.4.1.5951.4.1.1.83.6.26
- lsnNAT64UdpSessionsRate,1.3.6.1.4.1.5951.4.1.1.83.6.27

- lsnNAT64IcmpSessionsRate,1.3.6.1.4.1.5951.4.1.1.83.6.28
- ipConflictMacAddr,1.3.6.1.4.1.5951.4.1.10.2.66
- gslbSite,1.3.6.1.4.1.5951.4.1.10.2.67
- siteIP,1.3.6.1.4.1.5951.4.1.10.2.68
- lsnNAT64SubscrIPV6,1.3.6.1.4.1.5951.4.1.10.2.69

Deprecated SNMP OIDs

- ifTxCollisions,1.3.6.1.4.1.5951.4.1.1.54.1.14
- ifTxExcessCollisions,1.3.6.1.4.1.5951.4.1.1.54.1.15
- ifTxLateCollisions,1.3.6.1.4.1.5951.4.1.1.54.1.16
- ifTxMultiCollisionErrors,1.3.6.1.4.1.5951.4.1.1.54.1.17
- ifErrTxDeferred,1.3.6.1.4.1.5951.4.1.1.54.1.37

Deprecated features and functionalities

September 21, 2020

Some of the NetScaler functionalities are deprecated in 12.0 and in its subsequent releases. Citrix recommends that you do not use these features through the NetScaler command interface, NetScaler GUI, or Nitro automation. However, to ease the transition to new alternatives, the deprecated features are usable for a limited time and will be removed in future releases. Citrix recommends you to use the below alternatives for the following set of deprecated functionalities.

Deprecated features and Its alternatives in NetScaler 12.0

A list of NetScaler features and functionalities are deprecated. For complete information, You can refer to Deprecated List pdf.

Using NSPEPI conversion tool

You can use the nspepi tool to convert commands, expressions, and configurations. For more information, see Converting Expressions by Using NSPEPI Tool topic.
Here is a list of conversion from Classic expressions to Advanced expressions.

- ns_true -> true
- ns_false -> false
- REQ.HTTP -> HTTP.REQ
- RES.HTTP -> HTTP.RES

- HEADER “foo” -> HEADER(“foo”)
- CONTAINS ”bar” -> .CONTAINS(“bar”) [Note use of “.”.]
- REQ.IP -> CLIENT.IP
- RES.IP -> SERVER.IP
- SOURCEIP -> SRC
- DESTIP -> DST
- REQ.TCP -> CLIENT.TCP
- RES.TCP -> SERVER.TCP
- SOURCEPORT -> SRCPORT
- DESTPORT -> DSTPORT
- STATUSCODE -> STATUS
- REQ.SSL.CLIENT.CERT -> CLIENT.SSL.CLIENT_CERT

AppQoE Use Cases

When using AppQoE as an alternative for SureConnect feature, you can refer to the following use case configurations.

Use Case 1: To limit simultaneous download of huge files and display loading screen:

```
add appqoe action NS_RESP -respondWith NS -maxConn 50

add appqoe policy POL_DWN_PKG -rule “HTTP.REQ.URL.CONTAINS(\“tar.gz\”)” -action NS_RESP

bind lb vserver v1 -policyName POL_DWN_PKG -priority 8
<!--NeedCopy-->
```

Use Case 2: To limit simultaneous download of huge files and display loading screen with alternate content:

```
add appqoe action ACS_RESP -respondWith ACS -altContentsvcName s2 -altContentPath /netscaler/alt.html -maxConn 50

add appqoe policy POL_DWN_ACS -rule “HTTP.REQ.URL.CONTAINS(\“tar.gz\”)” -action ACS_RESP

bind lb vserver v1 -policyName POL_DWN_ACS -priority 9
<!--NeedCopy-->
```

Use Case 3: Custom loading screen:

```
import appqoCe customResp [http://<location>]|[local:<file>] simple_custom_resp
```

```

add appqoe action CUST_RESP -respondWith NS simple_custom_resp -maxConn 50

add appqoe policy POL_CRESP -rule “HTTP.REQ.URL.CONTAINS(\“tar.gz\”)” -action CUST_RESP

bind lb vserver v1 -policyName POL_CRESP -priority 10
<!--NeedCopy-->
```

AAA application traffic

September 21, 2020

Many companies restrict web site access to valid users only, and control the level of access permitted to each user. The AAA feature allows a site administrator to manage access controls with the NetScaler appliance instead of managing these controls separately for each application. Doing authentication on the appliance also permits sharing this information across all web sites within the same domain that are protected by the appliance.

The AAA feature supports authentication, authorization, and auditing for all application traffic. To use AAA, you must configure authentication virtual servers to handle the authentication process and traffic management virtual servers to handle the traffic to web applications that require authentication. You also configure your DNS to assign FQDNs to each virtual server. After configuring the virtual servers, you configure a user account for each user that will authenticate via the NetScaler appliance, and optionally you create groups and assign user accounts to groups. After creating user accounts and groups, you configure policies that tell the appliance how to authenticate users, which resources to allow users to access, and how to log user sessions. To put the policies into effect, you bind each policy globally, to a specific virtual server, or to the appropriate user accounts or groups. After configuring your policies, you customize user sessions by configuring session settings and binding your session policies to the traffic management virtual server. Finally, if your intranet uses client certs, you set up the client certificate configuration.

Before configuring AAA, you should be familiar with and understand how to configure load balancing, content switching, and SSL on the NetScaler appliance.

How AAA works

September 21, 2020

AAA provides security for a distributed internet environment by allowing any client with the proper credentials to connect securely to protected application servers from anywhere on the Internet. This feature incorporates the three security features of AAA. Authentication enables the NetScaler appliance to verify the client's credentials, either locally or with a third-party authentication server, and allow only approved users to access protected servers. Authorization enables the ADC to verify which content on a protected server it should allow each user to access. Auditing enables the ADC to keep a record of each user's activity on a protected server.

To understand how AAA works in a distributed environment, consider an organization with an intranet that its employees access in the office, at home, and when traveling. The content on the intranet is confidential and requires secure access. Any user who wants to access the intranet must have a valid user name and password. To meet these requirements, the ADC does the following:

- Redirects the user to the login page if the user accesses the intranet without having logged in.
- Collects the user's credentials, delivers them to the authentication server, and caches them in a directory that is accessible through LDAP.
- Verifies that the user is authorized to access specific intranet content before delivering the user's request to the application server.
- Maintains a session timeout after which users must authenticate again to regain access to the intranet. (You can configure the timeout.)
- Logs the user accesses, including invalid login attempts, in an audit log.

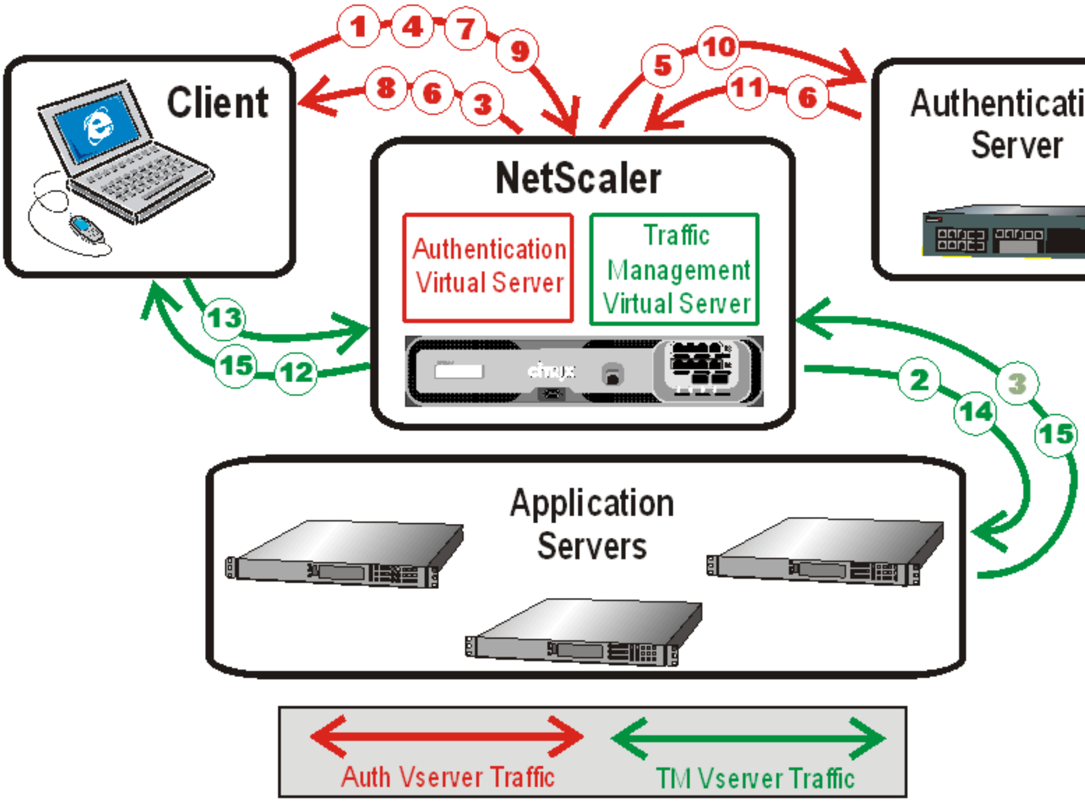
Authentication requires that several entities: the client, the
NetScaler appliance, the external authentication server if one is used, and the application server, respond to each other when prompted by performing a complex series of tasks in the correct order. If you are using an external authentication server, this process can be broken down into the following fifteen steps.

1. The client sends a GET request for a URL on the application server.
2. The NetScaler appliance's traffic management virtual server redirects the request to the application server.
3. The application server determines that the client has not been authenticated, and therefore sends an HTTP 200 OK response via the TM vserver to the client. The response contains a hidden script that causes the client to issue a POST request for /cgi/tm.
4. The client sends a POST request for /cgi/tm.
5. The NetScaler appliance's authentication virtual server redirects the request to the authentication server.
6. The authentication server creates an authentication session, sets and caches a cookie that consists of the initial URL and the domain of the traffic management virtual server, and then sends an HTTP 302 response via the authentication virtual server, redirecting the client to /vpn/index.html.
7. The client sends a GET request for /vpn/index.html.

8. The authentication virtual server redirects the client to the authentication server login page.
9. The client sends a GET request for the login page, enters credentials, and then sends a POST request with the credentials back to the login page.
10. The authentication virtual server redirects the POST request to the authentication server.
11. If the credentials are correct, the authentication server tells the authentication virtual server to log the client in and redirect the client to the URL that was in the initial GET request.
12. The authentication virtual server logs the client in and sends an HTTP 302 response that redirects the client to the initially requested URL.
13. The client sends a GET request for their initial URL.
14. The traffic management virtual server redirects the GET request to the application server.
15. The application server responds via the traffic management virtual server with the initial URL.

If you use local authentication, the process is similar, but the authentication virtual server handles all authentication tasks instead of forwarding connections to an external authentication server. The following figure illustrates the authentication process.

Figure 1. Authentication Process Traffic Flow


1
4
7
9
8
6
3
5
10
11
6
Client
Authentication Server
NetScaler
Authentication Virtual Server
Traffic Management Virtual Server
13
15
12
2
3
14
15
Application Servers
Auth Vserver Traffic
TM Vserver Traffic

When an authenticated client requests a resource, the ADC, before sending the request to the application server, checks the user and group policies associated with the client account, to verify that the client is authorized to access that resource. The ADC handles all authorization on protected application servers. You do not need to do any special configuration of your protected application servers.

AAA traffic management handles password changes for users by using the protocol-specific method for the authentication server. For most protocols, neither the user nor the administrator needs to do anything different than they would without AAA traffic management. Even when an LDAP authentication server is in use, and that server is part of a distributed network of LDAP servers with a single designated domain administration server, password changes are usually handled seamlessly. When an authenticated client of an LDAP server changes his or her password, the client sends a credential modify request to AAA traffic management, which forwards it to the LDAP server. If the user's LDAP server is also the domain administration server, that server responds appropriately and AAA traffic management then performs the requested password change. Otherwise, the LDAP server sends AAA traffic management an LDAP_REFERRAL response to the domain administration server. AAA traffic management follows the referral to the indicated domain administration server, authenticates to that server, and performs the password change on that server.

When configuring AAA traffic management with an LDAP authentication server, the system administrator must keep the following conditions and limitations in mind:

- AAA traffic management assumes that the domain administration server in the referral accepts the same bind credentials as the original server.
- AAA traffic management only follows LDAP referrals for password change operations. In other cases AAA traffic management refuses to follow the referral.
- AAA traffic management only follows one level of LDAP referrals. If the second LDAP server also returns a referral, AAA traffic management refuses to follow the second referral.

The ADC supports auditing of all states and status information, so you can see the details of what each user did while logged on, in chronological order. To provide this information, the appliance logs each event, as it occurs, either to a designated audit log file on the appliance or to a syslog server. Auditing requires configuring the appliance and any syslog server that you use.

Enabling AAA

September 21, 2020

To use the AAA - Application Traffic feature, you must enable it. You can configure AAA entities—such as the authentication and traffic management virtual servers—before you enable the AAA feature, but the entities will not function until the feature is enabled.

To enable AAA by using the command line interface

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

- enable ns feature AAA

- show ns feature

  Example

- `enable feature AAA<!--NeedCopy-->`

  Done

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

```
Feature                  Acronym              Status
-------                  -------              ------
1)  Web Logging            WL                   OFF
2)  Surge Protection       SP                   ON
 .
 .
 .
15)  AAA                  AAA                   ON
 .
 .
 .
23)    HTML Injection     HTMLInjection         ON
24)    NetScaler Push       push                OFF   Done
```

To enable AAA by using the configuration utility

1. Navigate to **System > Settings**.
2. In the details pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Authentication, Authorization and Auditing check box.
4. Click OK.

Setting up an authentication virtual server

December 16, 2020

All authentication requests are redirected by the traffic management virtual server (load balancing or content switching) to the authentication virtual sever. This virtual server processes the associated authentication policies and accordingly provides access to the application.

To set up an authentication virtual server by using the CLI

1. Enable the AAA feature.

ns-cli-prompt> **enable ns feature** AAA

2. Configure an authentication virtual server. It must be of type SSL and make sure to bind SSL certificate-key pair to the virtual server.

   ns-cli-prompt> **add authentication vserver <name> SSL <ipaddress> <port>**

   ns-cli-prompt> **bind ssl certkey <auth-vserver-name> <certkey>**

3. Specify the FQDN of the domain for the authentication virtual server.

   ns-cli-prompt> **set authentication vserver <name>** -**authenticationDomain** <FQDN>

4. Associate the authentication virtual server to the relevant traffic management virtual server.

> **Note:**
>
> The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

On the traffic management virtual server:

- Enable authentication.
- Specify the FQDN of the authentication virtual server as the authentication host of the traffic management virtual server.
- [Optional] Specify the authentication domain on the traffic management virtual server.

> **Note:**
>
> - If you do not configure the authentication domain, the appliance assigns an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication virtual server is **tm.xyz.bar.com**, the appliance assigns **xyz.bar.com** as the authentication domain.
> - If you have to set a domain wide cookie for an authentication domain, you must enable authentication profile on a load balancing virtual server.

For load balancing:
ns-cli-prompt> **set lb vserver <name>** -**authentication** ON -**authenticationhost** <FQDN> [-**authenticationdomain** <authdomain>]

For content switching:
ns-cli-prompt> **set cs vserver <name> <protocol> <IPAddress> <port>**

1. Verify that both the virtual servers are UP and configure correctly.

   ns-cli-prompt> **show authentication vserver** <name>

To set up an authentication virtual server by using the GUI

1. Enable the AAA feature.

Navigate to **System > Settings**, click **Configure Basic features**, and enable **Authentication, Authorization and Auditing**.

2. Configure the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and configure as required (refer to the configurations provided in the CLI procedure provided above).

3. Configure the traffic management virtual server for authentication.

   For load balancing:

   Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and configure the virtual server as required (refer to the configurations provided in the CLI procedure provided above).

   For content switching:

   Navigate to **Traffic Management > Content Switching > Virtual Servers**, and configure the virtual server as required (refer to the configurations provided in the CLI procedure provided above).

4. Verify the authentication setup.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and check the details of the relevant authentication virtual server.

Configuring the authentication virtual server

September 21, 2020

To configure AAA, first configure an authentication virtual server to handle authentication traffic. Next, bind an SSL certificate-key pair to the virtual server to enable it to handle SSL connections.
For additional information about configuring SSL and creating a certificate-key pair, see SSL certificates.

To configure an authentication virtual server by using the command line interface

To configure an authentication virtual server and verify the configuration, at the command prompt type the following commands in the order shown:

- `add authentication vserver <name> ssl <ipaddress><!--NeedCopy-->`
- `show authentication vserver <name><!--NeedCopy-->`
- `bind ssl certkey <certkeyName><!--NeedCopy-->`
- `show authentication vserver <name><!--NeedCopy-->`

- `set authentication vserver <name> -authenticationDomain <FQDN><!--NeedCopy-->`
- `show authentication vserver <name><!--NeedCopy-->`

Example

```
> add authentication vserver Auth-Vserver-2 SSL 10.102.29.77 443
 Done
> show authentication vserver Auth-Vserver-2
        Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
        State: DOWN[Certkey not bound]
        Client Idle Timeout: 180 sec
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
 Done
> bind ssl certkey Auth-Vserver-2 Auth-Cert-1
 Done
> show authentication vserver Auth-Vserver-2
        Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com
 Done
> set authentication vserver Auth-Vserver-2 -AuthenticationDomain myCompany.employee.com
 Done
> show authentication vserver Auth-Vserver-2
        Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
        State: DOWN[Certkey not bound]
        Client Idle Timeout: 180 sec
        Down state flush: DISABLED
        Disable Primary Vserver On Down : DISABLED
        Authentication : ON
        Current AAA Users: 0
        Authentication Domain: myCompany.employee.com
 Done
<!--NeedCopy-->
```

To configure an authentication virtual server by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers**.
2. In the details pane, do one of the following:
    - To create a new authentication virtual server, click **Add**.
    - To modify an existing authentication virtual server, select the virtual server, and then click **Edit**. The Configuration dialog opens with the Basic Settings area expanded.
3. Specify values for the parameters as follows (asterisk indicates a required parameter):
    - Name*—name (Cannot be changed for a previously created virtual server)
    - IP Address*—ipaddress
    - Domain*—authenticationDomain
    - Failed login timeout—failedLoginTimeout (Seconds allowed before login fails and user must start login process again.)
    - Max login attempts—maxLoginAttempts (Number of login attempts allowed before user is locked out)

    > Note: The authentication virtual server uses only the SSL protocol and port 443, so those options are greyed out. Any options that are not mentioned are not relevant and should be ignored.

4. Click **Continue** to display the Certificates area.
5. In the **Certificates** area, configure any SSL certificates you want to use with this virtual server.
    - To configure a CA certificate, click the arrow on the right of CA Certificate to display the CA Cert Key dialog box, select the certificate you want to bind to this virtual server, and click **Save**.
    - To configure a server certificate, click the arrow on the right of Server Certificate, and follow the same process as for CA certificate.
6. Click **Continue** to display the **Advanced Authentication Policies** area.
7. If you want to bind an advanced authentication policy to the virtual server, click the arrow on the right side of the line to display the **Authentication Policy** dialog box, choose the policy that you want to bind to the server, set the priority, and then click **OK**.
8. Click **Continue** to display the **Basic Authentication Policies** area.
9. If you want to create a basic authentication policy and bind it to the virtual server, click the plus sign to display the **Policies** dialog box, and follow the prompts to configure the policy and bind it to this virtual server.
10. Click **Continue** to display the 401-Based Virtual Servers area.

11. In the 401-Based Virtual Servers area, configure any load balancing or content switching virtual servers that you want to bind to this virtual server.
    - To bind a load balancing virtual server, click the arrow to the right of LB virtual server to display the LB Virtual Servers dialog box, and follow the prompts.
    - To bind a content switching virtual server, click the arrow to the right of CS virtual server to display the CS Virtual Servers dialog box, and follow the same process as to bind an LB virtual server.
12. If you want to create or configure a group, in the Groups area click the arrow to display the Groups dialog box, and follow the prompts.
13. Review your settings, and when you are finished, click **Done**. The dialog box closes. If you created a new authentication virtual server, it now appears in the **Configuration** window list.

Configuring a traffic management virtual server

September 21, 2020

After you have created and configured your authentication virtual server, you next create or configure a traffic management virtual server and associate your authentication virtual sever with it. You can use either a load balancing or content switching virtual server for a traffic management virtual server. For more information about creating and configuring either type of virtual server, see the *Citrix Traffic Management Guide* at Traffic Management.

> Note
>
> The FQDN of the traffic management virtual server must be in the same domain as the FQDN of the authentication virtual server for the domain session cookie to function correctly.

You configure a traffic management virtual server for AAA by enabling authentication and then assigning the FQDN of the authentication server to the traffic management virtual server. You can also configure the authentication domain on the traffic management virtual server at this time. If you do not configure this option, the NetScaler appliance assigns the traffic management virtual server an FQDN that consists of the FQDN of the authentication virtual server without the hostname portion. For example, if domain name of the authentication virtual server is tm.xyz.bar.com, the appliance assigns xyz.bar.com. as the authentication domain.

To configure a TM virtual server for AAA by using the command line interface

At the command prompt, type one of the following sets of commands to configure a TM virtual server and verify the configuration:

- `set lb vserver <name> -authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]<!--NeedCopy-->`
- `show lb vserver <name><!--NeedCopy-->`
- `set cs vserver <name> -authentication ON -authenticationhost <FQDN> [-authenticationdomain <authdomain>]<!--NeedCopy-->`
- `show cs vserver <name><!--NeedCopy-->`

  Example

```
> set lb vserver vs-cont-sw -Authentication ON -AuthenticationHost
   mywiki.index.com
 Done
> show lb vserver vs-cont-sw
        vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:15 2009 (+410 ms)
        Time since last state change: 5 days, 20:00:40.290
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
        Authentication: ON      Host: mywiki.index.com
 Done
<!--NeedCopy-->
```

To configure a TM virtual server for AAA by using the configuration utility

1. In the navigation pane, do one of the following.
    - Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
    - Navigate to **Traffic Management > Content Switching > Virtual Servers**
    - In the details pane, select the virtual server on which you want to enable authentication, and then click **Edit**.
    - In the Domain text box, type the authentication domain.
    - In the **Advanced** menu on the right, select **Authentication**.
    - Choose either **Form Based Authentication** or **401 Based Authentication**, and fill in the Authentication information.

  - For Form Based Authentication, enter the Authentication FQDN (the fully-qualified domain name of the authentication server), the Authentication VServer (the IP address of the authentication virtual server), and the Authentication Profile (the profile to use for authentication).
  - For 401 Based Authentication, enter the Authentication VServer and the Authentication Profile only.
- Click **OK**. A message appears in the status bar, stating that the vserver has been configured successfully.

Simplified login protocol support for AAA

The login protocol between AAA traffic management virtual servers and AAA virtual servers is simplified to use internal mechanisms as opposed to sending the encrypted data through query parameters. By leveraging this feature, the replay of requests is prevented.

Configuring DNS

September 21, 2020

For the domain session cookie used in the authentication process to function correctly, you must configure DNS to assign both the authentication and the traffic management virtual servers to FQDNs in the same domain. For information about how to the configure DNS address records, see Domain Name System.

Verifying your setup for AAA

September 21, 2020

After you configure authentication and traffic management virtual servers and before you create user accounts, you should verify that both virtual servers are configured correctly and are in the UP state.

To verify authentication virtual server setup by using the command line interface

At the command prompt, type the following command:

"‘show authentication vserver

```
> show authentication vserver Auth-Vserver-2
Auth-Vserver-2 (10.102.29.77:443) - SSL Type: CONTENT
State: UP
Client Idle Timeout: 180 sec
Down state flush: DISABLED
Disable Primary Vserver On Down : DISABLED
Authentication : ON
Current AAA Users: 0
Authentication Domain: myCompany.employee.com  Done
```

“‘

To verify your AAA virtual server setup by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers**.
2. Review the information in the AAA Virtual Servers pane to verify that your configuration is correct and your authentication virtual server is accepting traffic. You can select a specific virtual server to view detailed information in the details pane.

Creating an authentication profile

September 21, 2020

When you want the same authentication settings to be used by multiple traffic management virtual servers, you can create an authentication profile which specifies the authentication virtual server, the authentication host, the authentication domain, and authentication level.

This authentication profile can be associated with the relevant traffic management virtual servers.

To configure an authentication profile by using the CLI

- Create the authentication profile and set the required parameters.

  For example, to create a profile with an authentication virtual server named “authVS”.

  ns-cli-prompt> **add authentication authnProfile** authProfile1 -authnVsName authVS -authenticationHost authnVS.example.com -authenticationDomain example.com -authenticationLevel

> **Note:**
>
> The authentication weight or level will depend on the virtual server to which the traffic will bound. A session that is created by authenticating against traffic management virtual

> server at given level cannot be used to access traffic management virtual server at a higher level.

- Bind the authentication profile to the relevant traffic management virtual servers.

  For example, to bind authProfile1 to a load balancing virtual server named "vserver1".

  ns-cli-prompt> **set lb vserver** vserver1 -authnProfile authProfile1

To configure an authentication profile by using the GUI

In the **Configuration** tab, navigate to **Security > AAA - Application Traffic > Authentication Profile**, and configure the authentication profile as required.

Configuring users and groups

September 21, 2020

After configuring the AAA basic setup, you create users and groups. You first create a user account for each person who will authenticate via the NetScaler appliance. If you are using local authentication controlled by the NetScaler appliance itself, you create local user accounts and assign passwords to each of those accounts.

You also create user accounts on the NetScaler appliance if you are using an external authentication server. In this case, however, each user account must exactly match an account for that user on the external authentication server, and you do not assign passwords to the user accounts that you create on the NetScaler appliance. The external authentication server manages the passwords for users that authenticate with the external authentication server.

If you are using an external authentication server, you can still create local user accounts on the NetScaler appliance if, for example, you want to allow temporary users (such as visitors) to log in but do not want to create entries for those users on the authentication server. You assign a password to each local user account, just as you would if you were using local authentication for all user accounts.

Each user account must be bound to policies for authentication and authorization. To simplify this task, you can create one or more groups and assign user accounts to them. You can then bind policies to groups instead of individual user accounts.

To create a local AAA user account by using the command line interface

At the command prompt, type the following commands to create a local AAA user account and verify the configuration:

- `add aaa user <username> [-password <password>]<!--NeedCopy-->`

- `show aaa user<!--NeedCopy-->`

  Example

```
> add aaa user user-2 -password emptybag
 Done
> show aaa user
1) UserName: user-1
2) UserName: user-2
 Done
<!--NeedCopy-->
```

To change the password for an existing AAA local user account by using the command line interface

At the command prompt, type the following command and, when prompted, type the new password:

```
set aaa user <username>
```

Example

```
> set aaa user user-2
Enter password:
Done
<!--NeedCopy-->
```

To configure AAA local users by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Users**
2. In the details pane, do one of the following:
   - To create a new user account, click **Add**.
   - To modify an existing user account, select the user account, and then click **Open**.
3. In the **Create AAA User** dialog box, in the **User Name** text box, type a name for the user.
4. If creating a locally authenticated user account, clear the **External Authentication** check box and provide a local password that the user will use to log on
5. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that the user has been configured successfully.

To create AAA local groups and add users to them by using the command line interface

At the command prompt, type the following commands. Type the first command one time, and type the second command once for each user:

- `add aaa group <groupname><!--NeedCopy-->`
- “‘show aaa group

  ```
  **Example**
  ```

  > add aaa group group-2
  > Done
  > show aaa group
  > 1) GroupName: group-1
  > 2) GroupName: group-2
  > Done
  > “‘

- “‘bind aaa group -username

  ```
  **Example**
  ```

  > bind aaa group group-2 -username user-2
  > Done
  > show aaa group group-2
  > GroupName: group-2

  ```
              UserName: user-2    Done  <!--NeedCopy--> ```
  ```

To remove users from an AAA group by using the command line interface

At the command prompt, unbind users from the group by typing the following command once for each user account that is bound to the group:

- “‘unbind aaa group -username

  ```
  **Example**
  ```

  unbind aaa group group-hr -username user-hr-1
  Done
  “‘

To remove an AAA group by using the command line interface

First remove all users from the group. Then, at the command prompt, type the following command to remove an AAA group and verify the configuration:

- `rm aaa group <groupname><!--NeedCopy-->`

- “‘show aaa group

  ```
  **Example**
  ```

  ```
  rm aaa group group-hr
  Done
  show aaa group
  1) GroupName: group-1
  2) GroupName: group-finance
  Done
  ```

  “‘

To configure AAA local groups and add users to them by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Groups**
2. In the details pane, do one of the following:
   - To create a new group, click **Add**.
   - To modify an existing group, select the group, and then click **Edit**.
3. If you are creating a new group, in the **Create AAA Group** dialog box, in the **Group Name** text box, type a name for the group.
4. In the **Advanced** area to the right, click **AAA Users**.
   - To add a user to the group, select the user, and then click **Add**.
   - To remove a user from the group, select the user, and then click **Remove**.
   - To create a new user account and add it to the group, click the **Plus** icon, and then follow the instructions in “To configure AAA local users by using the configuration utility.”
5. Click **Create** or **OK**. The group that you created appears in the **AAA Groups** page.

Configuring AAA policies

September 21, 2020

After you set up your users and groups, you next configure authentication policies, authorization policies, and audit policies to define which users are allowed to access your intranet, which resources each user or group is allowed to access, and what level of detail AAA will preserve in the audit logs. An authentication policy defines the type of authentication to apply when a user attempts to log on. If external authentication is used, the policy also specifies the external authentication server. Authorization policies specify the network resources that users and groups can access after they log on. Auditing policies define the audit log type and location.

You must bind each policy to put it into effect. You bind authentication policies to authentication virtual servers, authorization policies to one or more user accounts or groups, and auditing policies both globally and to one or more user accounts or groups.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler appliance operating system, policy priorities work in reverse order: the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The AAA feature implements only the first of each type of policy that a request matches, not any additional policies of that type that a request might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind the policies. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler appliance, see the *Citrix Traffic Management Guide* at Traffic Management.

Authentication policies

September 21, 2020

The NetScaler appliance can authenticate users with local user accounts or by using an external authentication server. The appliance supports the following authentication types:

- **LOCAL**

  Authenticates to the NetScaler appliance by using a password, without reference to an external authentication server. User data is stored locally on the NetScaler appliance.

- **RADIUS**

  Authenticate to an external Radius server.

- **LDAP**

  Authenticates to an external LDAP authentication server.

- **TACACS**

  Authenticates to an external Terminal Access Controller Access-Control System (TACACS) authentication server.

After a user authenticates to a TACACS server, the NetScaler appliance connects to the same TACACS server for all subsequent authorizations. When a primary TACACS server is unavailable, this feature prevents delays while the ADC waits for the first TACACS server to time out before resending the authorization request to the second TACACS server.

> Note
>
> When authenticating through a TACACS server, AAA traffic management logs only successfully executed TACACS commands, to prevent the logs from showing TACACS commands that were entered by users who were not authorized to execute them.

Starting from NetScaler 12.0 Build 57.x, the Terminal Access Controller Access-Control System (TACACS) is not blocking the AAA daemon while sending the TACACS request. The AAA daemon allows LDAP, and RADIUS authentication to proceed with the request. The TACACS authentication request resumes once TACACS server acknowledges the TACACS request.

- **CERT**

  Authenticates to the NetScaler appliance by using a client certificate, without reference to an external authentication server.

- **NEGOTIATE**

  Authenticates to a Kerberos authentication server. If there is an error in Kerberos authentication, NetScaler appliance uses NTLM authentication.

- **SAML**

  Authenticates to a server that supports the Security Assertion Markup Language (SAML).

- **SAML IDP**

  Configures the NetScaler appliance to serve as a Security Assertion Markup Language (SAML) Identity Provider (IdP).

- **WEB**

  Authenticates to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful.

An authentication policy is comprised of an expression and an action. Authentication policies use NetScaler appliance expressions.

After creating an authentication action and an authentication policy, bind it to an authentication virtual server and assign a priority to it. When binding it, also designate it as either a primary or a secondary policy. Primary policies are evaluated before secondary policies. In configurations that use both types of policy, primary policies are normally more specific policies while secondary policies are normally more general policies intended to handle authentication for any user accounts that do not meet the more specific criteria.

To add an authentication action by using the command line interface

If you do not use LOCAL authentication, you need to add an explicit authentication action. To do this, at the command prompt, type the following command:

"'add authentication tacacsAction -serverip [-serverPort][-authTimeout][…]

```
Example
```

> add authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -tacacsSecret "minotaur" -authorization OFF -accounting ON -auditFailedCmds OFF -defaultAuthenticationGroup "users"
> Done

~~~
## To configure an authentication action by using the command line
    interface

To configure an existing authentication action, at the command prompt,
    type the following command:

```set authentication tacacsAction <name> -serverip <IP> [-serverPort <
    port>][-authTimeout <positive_integer>][ ... ]<!--NeedCopy-->
~~~

Example

```
> set authentication tacacsaction Authn-Act-1 -serverip 10.218.24.65 -
    serverport 1812                    -authtimeout 15 -tacacsSecret "
    minotaur" -authorization OFF -accounting ON -auditFailedCmds OFF -
    defaultAuthenticationGroup "users"
Done

<!--NeedCopy-->
```

To remove an authentication action by using the command line interface

To remove an existing RADIUS action, at the command prompt, type the following command:

"'rm authentication radiusAction

```
Example
```

> rm authentication tacacsaction Authn-Act-1
> Done

````
## To configure an authentication server by using the configuration utility

> Note
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1.  Navigate to **Security > AAA - Application Traffic > Policies > Authentication**.
1.  In the details pane, on the **Servers** tab, do one of the following:
    -  To create a new authentication server, click **Add**.
    -  To modify an existing authentication server, select the server, and then click **Open**.
1.  In the **Create Authentication Server** or **Configure Authentication Server** dialog box, type or select values for the parameters.
    -  Name\* — radiusActionName (Cannot be changed for a previously configured action)
    -  Authentication Type\* — authtype (Set to RADIUS, cannot be changed)
    -  IP Address\* — serverip </IP>
    -  IPV6\* — Select the checkbox if the server IP is an IPv6 IP. (No command line equivalent.)
    -  Port\* — serverPort
    -  Time-out (seconds)\* — authTimeout
1.  Click **Create** or **OK**, and then click **Close**. The policy that you created appears in the **Authentication Policies** and **Servers** page.

## To create and bind an authentication policy by using the command line interface

At the command prompt, type the following commands in the order shown to create and bind an authentication policy and verify the configuration:

- ```add authentication negotiatePolicy <name> <rule> <reqAction><!--NeedCopy-->
````

- “‘show authentication localPolicy
- “‘bind authentication vserver -policy [-priority][-secondary]]

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

Example

```
  > add authentication localPolicy Authn-Pol-1 ns_true   Done
  > show authentication localPolicy
  1)      Name: Authn-Pol-1       Rule: ns_true          Request action
      : LOCAL    Done
  > bind authentication vserver Auth-Vserver-2 -policy Authn-Pol-1
  Done
  > show authentication vserver Auth-Vserver-2          Auth-Vserver-2
      (10.102.29.77:443) - SSL Type: CONTENT State: UP Client Idle
      Timeout: 180 sec Down state flush: DISABLED          Disable
      Primary Vserver On Down : DISABLED         Authentication : ON
               Current AAA Users: 0          Authentication Domain:
      myCompany.employee.com
  1)  Primary authentication policy name: Authn-Pol-1 Priority: 0
  Done

<!--NeedCopy-->
```

To modify an existing authentication policy by using the command line interface

At the command prompt, type the following commands to modify an existing authentication policy:

"'set authentication localPolicy [-reqaction]

```
Example
```

> set authentication localPolicy Authn-Pol-1 'ns_true' Done
> "'

To remove an authentication policy by using the command line interface

At the command prompt, type the following command to remove an authentication policy:

"'rm authentication localPolicy

```
Example
```

> rm authentication localPolicy Authn-Pol-1
>
> Done

"'

To configure and bind authentication policies by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication**, and then select the type of policy that you want to create.
2. In the details pane, on the **Policies** tab, do one of the following:
    - To create a new policy, click **Add**.
    - To modify an existing policy, select the action, and then click **Edit**.
3. In the Create Authentication Policy or Configure Authentication Policy dialog, type or select values for the parameters.
    - Name*—policyname (Cannot be changed for a previously configured action)
    - Authentication Type*—authtype
    - Server*—authVsName
    - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
4. Click **Create** or **OK**. The policy that you created appears in the Policies page.
5. Click the **Servers** tab, and in the details pane do one of the following:
    - To use an existing server, select it, and then click .
    - To create a new server, click Add, and follow the instructions.
6. If you want to designate this policy as a secondary authentication policy, on the Authentication tab, click Secondary. If you want to designate this policy as a primary authentication policy, skip this step.
7. Click **Insert Policy**.
8. Choose the policy you want to bind to the authentication virtual server from the drop-down list.
9. In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
10. Click **OK**. A message appears in the status bar, stating that the policy has been configured successfully.

LDAP authentication policies

September 21, 2020

As with other types of authentication policies, a Lightweight Directory Access Protocol (LDAP) authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. In addition to standard authentication functions, LDAP can search other active directory (AD) servers for user accounts for users that do not exist locally. This function is called referral support or referral chasing.

Normally you configure the NetScaler appliance to use the IP address of the authentication server during authentication. With LDAP authentication servers, you can also configure the ADC to use the FQDN of the LDAP server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you use the **serverName** parameter instead of the **serverIP** parameter, and substitute the server's FQDN for its IP address.

Before you decide whether to configure the ADC to use the IP or the FQDN of your LDAP server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

LDAP referral support is disabled by default and cannot be enabled globally. It must be explicitly enabled for each LDAP action. You must also make sure that the AD server accepts the same binddn credentials that are used with the referring (GC) server. To enable referral support, you configure an LDAP action to follow referrals, and specify the maximum number of referrals to follow.

If referral support is enabled, and the NetScaler appliance receives an LDAP_REFERRAL response to a request, AAA follows the referral to the active directory (AD) server contained in the referral and performs the update on that server. First, AAA looks up the referral server in DNS, and connects to that server. If the referral policy requires SSL/TLS, it connects via SSL/TLS. It then binds to the new server with the binddn credentials that it used with the previous server, and performs the operation which generated the referral. This feature is transparent to the user.

> Note
>
> These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

For more information about setting up authentication policies in general, see Authentication Policies. For more information about NetScaler appliance expressions, which are used in the policy rule, see Policies and Expressions.

To enable LDAP referral support by using the command line interface

At the command prompt, type the following commands:

- `set authentication ldapAction <name> -followReferrals ON<!--NeedCopy-->`
- `set authentication ldapAction <name> -maxLDAPReferrals <integer><!--NeedCopy-->`

Example

```
> set authentication ldapAction ldapAction-1 -followReferrals ON
> set authentication ldapAction ldapAction-1 -maxLDAPReferrals 2
<!--NeedCopy-->
```

To enable LDAP referral support by using the configuration utility

> Note
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > LDAP**.
2. In the details pane, on the **Servers** tab, select the LDAP server that you want to configure, and then click **Edit**.
3. In the **Configure Authentication Server** dialog, scroll down to the **Referrals** check box, and select it.
4. In the **Maximum Referral Level** text box, type the maximum number of referrals to allow.
5. Click **OK**, and then click **Close**.

Negotiate authentication policies

September 21, 2020

As with other types of authentication policies, a Negotiate authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

In addition to standard authentication functions, the Negotiate Action command can now extract user information from a keytab file instead of requiring you to enter that information manually. If a keytab has more than one SPN, AAA selects the correct SPN. You can configure this feature at the command line, or by using the configuration utility.

Note: These instructions assume that you are already familiar with the LDAP protocol and have already configured your chosen LDAP authentication server.

To configure AAA to extract user information from a keytab file by using the command line interface

At the command prompt, type the appropriate command:

- `add authentication negotiateAction <name> [-keytab <string>]<!--NeedCopy -->`
- `set authentication negotiateAction <name> [-keytab <string>]<!--NeedCopy -->`

Example

```
> set authentication negotiateAction negotiateAction-1 -keytab keytab-1

<!--NeedCopy-->
```

To configure AAA to extract user information from a keytab file by using the configuration utility

> Note
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > Negotiate**.
2. In the details pane, on the **Servers** tab, do one of the following:
    - If you want to create a new **Negotiate** action, click **Add**.
    - If you want to modify an existing **Negotiate** action, in the data pane select the action, and then click **Edit**.
3. If you are creating a new **Negotiate** action, in the **Name** text box, type a name for your new action. The name can be from one to 127 characters in length and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing Negotiate action, skip this step. The name is read-only; you cannot change it.
4. Under **Negotiate**, if the Use Keytab file check box is not already checked, check it.
5. In the Keytab file path text box, type the full path and filename of the keytab file that you want to use.
6. In the Default authentication group text box, type the authentication group that you want to set as default for this user.
7. Click **Create** or **OK** to save your changes.

RADIUS authentication policies

October 5, 2020

As with other types of authentication policies, a Remote Authentication Dial In User Service (RADIUS) authentication policy comprises of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy. However, setting up a RADIUS authentication policy has certain special requirements that are described in the following section.

Normally you configure the NetScaler appliance to use the IP address of the authentication server during authentication. With RADIUS authentication servers, you can now configure the ADC to use the FQDN of the RADIUS server instead of its IP address to authenticate users. Using an FQDN can simplify an otherwise much more complex AAA configuration in environments where the authentication server might be at any of several IP addresses, but always uses a single FQDN. To configure authentication by using a server's FQDN instead of its IP address, you follow the normal configuration process except when creating the authentication action. When creating the action, you substitute the **serverName** parameter for the **serverIP** parameter.

Before you decide whether to configure the NetScaler appliance to use the IP or the FQDN of your RADIUS server to authenticate users, consider that configuring AAA to authenticate to an FQDN instead of an IP address adds an extra step to the authentication process. Each time the ADC authenticates a user, it must resolve the FQDN. If a great many users attempt to authenticate simultaneously, the resulting DNS lookups might slow the authentication process.

> **Note**
>
> These instructions assume that you are already familiar with the RADIUS protocol and have already configured your chosen RADIUS authentication server.

For more information about setting up authentication policies in general, see Authentication Policies. For more information about NetScaler appliance expressions, which are used in the policy rule, see Policies and Expressions.

To add an authentication action for a RADIUS server by using the command line interface

If you authenticate to a RADIUS server, you need to add an explicit authentication action. At the command prompt, type the following command:

"'add authentication radiusAction [-serverip | -serverName]][-serverPort] [-authTimeout] {-radKey } [-radNASip (ENABLED | DISABLED)][-radNASid] [-radVendorID][-radAttributeType][-radGroupsPrefix] [-radGroupSeparator][-passEncoding][-ipVendorID] [-ipAttributeType][-accounting (ON | OFF)][-pwdVendorID [-pwdAttributeType]] [-defaultAuthenticationGroup] [-callingstationid (ENABLED | DISABLED)]

```
The following example adds a RADIUS authentication action named **Authn
    -Act-1**, with the server IP **10.218.24.65**, the server port
    **1812**, the authentication timeout **15** minutes, the RADIUS key
    **WareTheLorax**, NAS IP disabled, and NAS ID **NAS1**.
```

> add authentication radiusaction Authn-Act-1 -serverip 10.218.24.65 -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
> Done

```
The following example adds the same RADIUS authentication action, but
    using the server FQDN **rad01.example.com** instead of the IP.
```

> add authentication radiusaction Authn-Act-1 -serverName rad01.example.com -serverport 1812 -authtimeout 15 -radkey WareTheLorax -radNASip DISABLED -radNASid NAS1
> Done

```
RADIUS authentication might fail if both authentication and accounting
    actions are configured on the same server port in “radiusAction”
    command.

To overcome the preceding conditions, Citrix recommends you to use
    different commands for RADIUS authentication and accounting.

-  For RADIUS authentication, configure the **authservRetry** parameter
     with default value 3.

    -  ```set authentication radiusAction Authn-Act-1 -serverip
       10.218.24.65 - serverport 1812 -authservRetry 3<!--NeedCopy-->
```

- For RADIUS accounting, configure the **authservRetry** parameter with the value 1.
  - `set authentication radiusAction Authn-Act-1 -serverip 10.218.24.65 -serverport 1813 -authservRetry 1<!--NeedCopy-->`

To configure an authentication action for an external RADIUS server by using the command line

To configure an existing RADIUS action, at the command prompt, type the following command:

“‘set authentication radiusAction [-serverip | -serverName]][-serverPort] [-authTimeout] {-radKey } [-radNASip (ENABLED | DISABLED)][-radNASid] [-radVendorID][-radAttributeType][-radGroupsPrefix] [-radGroupSeparator][-passEncoding][-ipVendorID] [-ipAttributeType][-accounting (ON | OFF)][-pwdVendorID [-pwdAttributeType]] [-defaultAuthenticationGroup] [-callingstationid (ENABLED | DISABLED)]

```
## To remove an authentication action for an external RADIUS server by
    using the command line interface

To remove an existing RADIUS action, at the command prompt, type the
    following command:
```

```rm authentication radiusAction <name><!--NeedCopy-->
```

Example

```
> rm authentication radiusaction Authn-Act-1
Done

<!--NeedCopy-->
```

To configure a RADIUS server by using the configuration utility

> Note
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Radius**
2. In the details pane, on the **Servers** tab, do one of the following:
    - To create a new RADIUS server, click **Add**.
    - To modify an existing RADIUS server, select the server, and then click **Edit**.
3. In the **Create Authentication RADIUS Server** or **Configure Authentication RADIUS Server** dialog, type or select the values for the parameters. To fill out parameters that appear beneath Send Calling Station ID, expand Details.
    - Name*—radiusActionName (Cannot be changed for a previously configured action)
    - Authentication Type*—authtype (Set to RADIUS, cannot be changed)
    - Server Name / IP Address*—Choose either Server Name or Server IP
        - Server Name*—serverName <FQDN>
        - IP Address*—serverIp <IP> If the server is assigned an IPv6 IP address, select the IPv6 check box.
    - Port*—serverPort
    - Time-out (seconds)*—authTimeout
    - Secret Key*—radKey (RADIUS shared secret.)
    - Confirm Secret Key*—Type the RADIUS shared secret a second time. (No command line equivalent.)
    - Send Calling Station ID—callingstationid
    - Group Vendor Identifier—radVendorID
    - Group Attribute Type—radAttributeType
    - IP Address Vendor Identifier—ipVendorID
    - pwdVendorID—pwdVendorID
    - Password Encoding—passEncoding
    - Default Authentication Group—defaultAuthenticationGroup
    - NAS ID—radNASid

  - Enable NAS IP address extraction—radNASip
  - Group Prefix—radGroupsPrefix
  - Group Separator—radGroupSeparator
  - IP Address Attribute Type—ipAttributeType
  - Password Attribute Type—pwdAttributeType
  - Accounting—accounting

4. Click **Create** or **OK**. The policy that you created appears in the Servers page.

SAML authentication policies

September 21, 2020

For information on NetScaler appliance as a SAML SP and IdP, see SAML Authentication.

Web authentication policies

September 21, 2020

AAA is now able to authenticate a user to a web server, providing the credentials that the web server requires in an HTTP request and analyzing the web server response to determine that user authentication was successful. As with other types of authentication policies, a Web authentication policy is comprised of an expression and an action. After creating an authentication policy, you bind it to an authentication virtual server and assign a priority to it. When binding it, you also designate it as either a primary or a secondary policy.

To set up web-based authentication with a specific web server, first you create a web authentication action. Since authentication to web servers does not use a rigid format, you must specify exactly which information the web server requires and in which format when creating the action. To do this, you create an expression in NetScaler appliance default syntax that contains the following items:

- **Server IP**—The IP address of the authentication Web server.
- **Server Port**—The port of the authentication Web server.
- **Authentication Rule**—An expression in NetScaler appliance default syntax that contains the user's credentials in the format that the Web server expects.
- **Scheme**—HTTP (for unencrypted web authentication) or HTTPS (for encrypted web authentication).
- **Success Rule**—An expression in NetScaler appliance default syntax that matches the web server response string that signifies that the user authenticated successfully.

For all other parameters, follow the normal rules for the add authentication action command.

Next you create a policy associated with that action. The policy is similar to an LDAP policy, and like LDAP policies uses NetScaler appliance syntax.

> Note
>
> These instructions assume that you are already familiar with the authentication requirements of the web server(s) to which you want to authenticate, and have already configured the web authentication server.

To configure a Web authentication action by using the command line interface

To create a web authentication action at the command line, at the command line type the following command:

“‘add authentication webAuthAction -serverIP <ip_addr|ipv6_addr|*> -serverPort <port|*> [-fullReqExpr] -scheme (http | https) -successRule [-defaultAuthenticationGroup][-Attribute1][-Attribute2] [-Attribute3][-Attribute4] [-Attribute5][-Attribute6] [-Attribute7][-Attribute8] [-Attribute9][-Attribute10] [-Attribute11][-Attribute12] [-Attribute13][-Attribute14] [-Attribute15][-Attribute16]

```
Example
```

> add authentication webAuthAction webauth1 -serverIP 10.214.56.31 -serverPort 80 -

“‘

To configure a Web authentication action by using the configuration utility

> **Note**
>
> In the configuration utility, the term server is used instead of action, but refers to the same task.

1. Navigate to **Security > AAA - Application Traffic > Policies > LDAP**.
2. In the details pane, on the **Servers** tab, do one of the following:
    - If you want to create a new web authentication action, click **Add**.
    - If you want to modify an existing web authentication action, in the data pane select the action, and then click **Edit**.
3. If you are creating a new web authentication action, in the **Create Authentication Web** server dialog box, **Name** text box, type a name for the new web authentication action. The name can be from one to 127 characters in length, and can consist of upper- and lowercase letters, numbers, and the hyphen (-) and underscore (_) characters. If you are modifying an existing web authentication action, skip this step. The name is read-only; you cannot change it.
4. In the **Web Server IP Address** text box, type the IPv4 or IPv6 IP address of the authentication web server. If the address is an IPv6 IP address, select the IPv6 check box first.

5. In the Port text box, type the port number on which the web server accepts connections.
6. Select **HTTP** or **HTTPS** in the **Protocol** drop-down list.
7. In the HTTP Request Expression text area, type a PCRE-format regular expression that creates the web server request that contains the user's credentials in the exact format expected by the authentication web server.
8. In the Expression to validate the Authentication text area, type a NetScaler appliance default syntax expression that describes the information in the web server response that indicates that user authentication was successful.
9. Fill out the remaining fields as described in the general authentication action documentation.
10. Click **OK**.

Configuring advanced authentication policies

September 21, 2020

If you know exactly how you want an authentication policy to be configured, you can use the advanced authentication policy dialog to create the policy quickly.

To configure an advanced authentication policy by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies**, and then select Policy.
2. In the details pane do one of the following:
    - To create a new policy, click **Add**.
    - To modify an existing policy, select the policy, and then click **Edit**.
3. In the **Create Authentication Policy** or **Configure Authentication Policy** dialog box, type or select values for the parameters.
    - Name*—The policy name. Cannot be changed for a previously configured policy.
    - Action Type*—The policy type: Cert, Negotiate, LDAP, RADIUS, SAML, SAMLIDP, TACACS, or WEBAUTH.
    - Action*—The authentication action (profile) to associate with the policy. You can choose an existing authentication action, or click the plus and create a new action of the proper type.
    - Log Action—The audit action to associate with the policy. You can choose an existing audit action, or click the plus and create a new action.
    - Expression*—The rule that selects connections to which you want to apply the action that you specified. The rule can be simple ("true" selects all traffic) or complex. You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression window, and then by typing your expression directly into the expression

    text area, or by clicking Add to open Add Expression dialog box and using the drop-down lists in it to construct your expression.)
    - Comment—You can type a comment that describes the type of traffic that this authentication policy will apply to. Optional.
4. Click **Create** or **OK**, and then click **Close**. If you created a policy, that policy appears in the Authentication Policies and Servers page.

Authorizing user access to application resources

September 21, 2020

You can control the resources that an authenticated user can access within an application.

To do this, associate an authorization policy to each of the users, either individually or by associating the policy to a group of users. The authorization policy must specify the following:

- **Rule.** The resource to which access must be authorized. This can be specified by using basic or advanced expressions.
- **Action.** Whether access to the resource must be allowed or denied.

By default, access to all resources within an application is **DENIED** to all users. However, you can change this default authorization action to **ALLOW** access to all users (by setting the session parameters in session profile or by setting the global session parameters).

> **Warning**
>
> For optimum security, Citrix recommends that you do not to change the default authorization action from DENY to ALLOW. Instead, it is advised to create specific authorization policies for users who need access to specific resources.

To configure authorization by using the CLI

1. Configure the authorization policy.

   ns-cli-prompt> **add authorization policy <name> <rule> <action>**

2. Associate the policy with the appropriate user or group.

    - Bind the policy to a specific user.
      ns-cli-prompt> **bind aaa user <username>** -**policy** <policyname>

    - Bind the policy to a specific group.
      ns-cli-prompt> **bind aaa group <groupName> -policy** <policyname>

To configure authorization by using the GUI (Configuration tab)

1. Create the authorization policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authorization**, click **Add** and then define the policy as required.

2. Associate the policy with the appropriate user or group.

   Navigate to **Security > AAA - Application Traffic > Users** or **Groups**, and edit the relevant user or group to associate it with the authorization policy.

Sample authorization configurations

Here are some example configurations to authorize user access to some application resources. Note that these are CLI commands. You can do similar configurations using the GUI, although you must not enclose the expression within quotes (").

- `add authorization policy authzpol1 "HTTP.REQ.URL.SUFFIX.EQ(\"gif\")" ALLOW<!--NeedCopy-->`
- `bind aaa user user1 -policy authzpol1<!--NeedCopy-->`
- `add authorization policy authzpol2 "HTTP.REQ.URL.SUFFIX.EQ(\"png\")" DENY<!--NeedCopy-->`
- `bind aaa group group1 -policy authzpol2<!--NeedCopy-->`

Auditing authenticated sessions

September 21, 2020

You can configure the NetScaler appliance to keep a log of all the events that are triggered in an authenticated session. Using this information, you can audit state and status information, to see the history for users in chronological order.

To do this, define an audit policy that specifies the following:

- **Log type.** The logs can be stored remotely (syslog) or locally on the NetScaler appliance (nslog).
- **Rule.** The conditions on which the logs are stored.
- **Action.** Details of the log server and other details for creating the log entries.

This audit policy can be configured at different levels: user-level, group-level, AAA virtual server, and global system level. The policies configured at the user-level have the highest priority.

> **Note**
>
> This topic details steps for using syslog. Make necessary changes to use nslog.

To configure syslog auditing by using the CLI

1. Configure the audit server with the relevant log settings.

   ns-cli-prompt> **add audit syslogAction <name> <serverIP> …**

2. Configure the audit policy by associating the audit server.

   ns-cli-prompt> **add audit syslogPolicy <name> <rule> <action>**

3. Associate the audit policy with one of the following entities:

   - Bind the policy to a specific user.

     ns-cli-prompt> **bind aaa user <userName>**-**policy** <policyname> …

   - Bind the policy to a specific group.

     ns-cli-prompt> **bind aaa group <groupName>**-**policy** <policyname> …

   - Bind the policy to a AAA virtual server.

     ns-cli-prompt> **bind authentication vserver <name>** -**policy** <policyname> …

   - Bind the policy globally to the NetScaler appliance.

     ns-cli-prompt> **bind tm global** -**policyName** <policyname> …

To configure syslog auditing by using the GUI (Configuration tab)

1. Configure the audit server and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog**, and configure the server and the policy in the relevant tabs.

2. Associate the policy with one of the following:

   - Bind the policy to a specific user.

     Navigate to **Security > AAA - Application Traffic > Users**, and associate the authorization policy with the relevant user.

   - Bind the policy to a specific group.

     Navigate to **Security > AAA - Application Traffic > Groups**, and associate the authorization policy with the relevant group.

- Bind the policy to a AAA virtual server.

  Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the authorization policy with the relevant virtual server.

- Bind the policy globally to the NetScaler appliance.

  Navigate to **Security > AAA - Application Traffic > Policies > Auditing > Syslog** or **Nslog**, select the authorization policy, and click **Action > Global Bindings** to bind the policy globally.

Session settings

September 21, 2020

After you configure your AAA profiles, you configure session settings to customize your user sessions. The session settings are:

- **The session timeout.**

  Controls the period after which the user is automatically disconnected and must authenticate again to access your intranet.

- **The default authorization setting.**

  Determines whether the
  NetScaler appliance will by default allow or deny access to content for which there is no specific authorization policy.

- **The single sign-on setting.**

  Determines whether the
  NetScaler appliance will log users onto all web applications automatically after they authenticate, or will pass users to the web application logon page to authenticate for each application.

- **The credential index setting.**

  Determines whether the
  NetScaler appliance will use primary or the secondary authentication credentials for single signon.

To configure the session settings, you can take one of two approaches. If you want different settings for different user accounts or groups, you create a profile for each user account or group for which you want to configure custom sessions settings. You also create policies to select the connections to which to apply particular profiles, and you bind the policies to users or groups. You can also bind a policy to the authentication virtual server that handles the traffic to which you want to apply the profile.

If you want the same settings for all sessions, or if you want to customize the default settings for sessions that do not have specific profiles and policies configured, you can simply configure the global session settings.

Session profiles

September 21, 2020

To customize your user sessions, you first create a session profile. The session profile allows you to override global settings for any of the session parameters.

> **Note**
>
> The terms "session profile" and "session action" mean the same thing.

To create a session profile by using the command line interface

At the command prompt, type the following commands to create a session profile and verify the configuration:

- `add tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)][-SSO (ON | OFF)][-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>][-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]<!--NeedCopy-->`
- `show tm sessionAction <name><!--NeedCopy-->`

Example

```
> add tm sessionAction session-profile -sessTimeout 30 -defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

To modify a session profile by using the command line interface

At the command prompt, type the following commands to modify a session profile and verify the configuration:

- `set tm sessionAction <name> [-sessTimeout <mins>] [-defaultAuthorizationAction (ALLOW | DENY)][-SSO (ON | OFF)][-ssoCredential (PRIMARY | SECONDARY)] [-ssoDomain <string>][-httpOnlyCookie (YES | NO)] [-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity <minutes>]<!--NeedCopy-->`
- `show tm sessionAction<!--NeedCopy-->`

Example

```
> set tm sessionAction session-profile -sessTimeout 30 -
   defaultAuthorization ALLOW
 Done
> show tm sessionAction session-profile
1)      Name: session-profile
        Authorization action : ALLOW
        Session timeout: 30 minutes
 Done
<!--NeedCopy-->
```

To remove a session profile by using the command line interface

At the command prompt, type the following command to remove a session profile:

`rm tm sessionAction <name><!--NeedCopy-->`

To configure session profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Session.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Session.**
3. In the details pane, click the **Profiles** tab.
4. On the **Profiles** tab, do one of the following:
   - To create a new session profile, click **Add.**
   - To modify an existing session profile, select the profile, and then click **Edit.**
5. In the Create TM Session Profile or Configure TM Session Profile dialog, type or select values for the parameters.
   - Name*—actionname (Cannot be changed for a previously configured session action.)
   - Session Time-out—sesstimeout
   - Single Signon to Web Applications—sso

   - Default Authorization Action—defaultAuthorizationAction
   - Credential Index—ssocredential
   - Single Sign-on Domain—ssoDomain
   - HTTPOnly Cookie—httpOnlyCookie
   - Enable Persistent Cookie—persistentCookie
   - Persistent Cookie Validity—persistentCookieValidity

6. Click **Create** or **OK**. The session profile that you created appears in the Session Policies and Profiles pane.

Session policies

September 21, 2020

After you create one or more session profiles, you create session policies and then bind the policies globally or to an authentication virtual server to put them into effect.

To create a session policy by using the command line interface

At the command prompt, type the following commands to create a session policy and verify the configuration:

```
-  add tm sessionPolicy <name> <rule> <action>
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

Example

```
> add tm sessionPolicy session-pol "URL == /*.png" session-profile
Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

To modify a session policy by using the command line interface

At the command prompt, type the following commands to modify a session policy and verify the configuration:

```
-  set tm sessionPolicy <name> [-rule <expression>] [-action <action>]
-  show tm sessionPolicy <name>
<!--NeedCopy-->
```

Example

```
> set tm sessionPolicy session-pol "URL == /*.png" session-profile
 Done
> show tm sessionPolicy session-pol
1)      Name: session-pol       Rule: URL == '/*.png'
        Action: session-profile
 Done
<!--NeedCopy-->
```

To globally bind a session policy by using the command line interface

At the command prompt, type the following commands to globally bind a session policy and verify the configuration:

"'bind tm global -policyName [-priority]

```
Example
```

> bind tm global -policyName session-pol
> Done

> show tm sessionPolicy session-pol
> 1) Name: session-pol Rule: URL == '/*.png'
> Action: session-profile
> Policy is bound to following entities
> 1) TM GLOBAL PRIORITY : 0
> Done

~~~
## To bind a session policy to an authentication virtual server by
   using the command line interface

At the command prompt, type the following command to bind a session
   policy to an authentication virtual and verify the configuration:

```bind authentication vserver <name> -policy <policyname> [-priority <
   priority>]<!--NeedCopy-->
~~~

Example

```
> bind authentication vserver auth-vserver-1 -policyName Session-Pol-1
    -priority 1000
Done
<!--NeedCopy-->
```

To unbind a session policy from an authentication virtual server by using the command line interface

At the command prompt, type the following commands to unbind a session policy from an authentication virtual server and verify the configuration:

"'unbind authentication vserver -policy

```
Example
```

> unbind authentication vserver auth-vserver-1 -policyName Session-Pol-1
> Done

~~~
## To unbind a globally bound session policy by using the command line
    interface

At the command prompt, type the following commands to unbind a globally
    -bound session policy:

```unbind tm global -policyName <policyname><!--NeedCopy-->
~~~

Example

```
> unbind tm global -policyName Session-Pol-1
Done
<!--NeedCopy-->
```

To remove a session policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type the following commands to remove a session policy and verify the configuration:

"'rm tm sessionPolicy

```
Example
```

```
rm tm sessionPolicy Session-Pol-1
Done
```

“‘

To configure and bind session policies by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Session**.
2. Navigate to **Security > AAA - Application Traffic > Policies > Session**.
3. In the details pane, on the **Policies** tab, do one of the following:
   - To create a new session policy, click **Add**.
   - To modify an existing session policy, select the policy, and then click **Edit**.
4. In the **Create Session Policy** or **Configure Session Policy** dialog, type or select values for the parameters.
   - Name*—policyname (Cannot be changed for a previously configured session policy.)
   - Request Profile*—actionname
   - Expression*—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking **Add** to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click **Create** or **OK**. The policy that you created appears in the details pane of the **Session Policies** and **Profiles** page.
6. To globally bind a session policy, in the details pane, select **Global Bindings** from the **Action** drop-down list, and fill in the dialog.
   - Select the name of the session policy you want to globally bind.
   - Click **OK**.
7. To bind a session policy to an authentication virtual server, in the navigation pane, click **Virtual Servers**, and add that policy to the policies list.
   - In the details pane, select the virtual server, and then click **Edit**.
   - In the **Advanced Selections** to the right of the detail area, click **Policies**.
   - Select a policy, or click the **plus** icon to add a policy.
   - In the **Priority** column to the left, modify the default priority as needed to ensure that the policy is evaluated in the proper order.
   - Click **OK**.
     A message appears in the status bar, stating that the policy has been configured successfully.

Global session settings

September 21, 2020

In addition to or instead of creating session profiles and policies, you can configure global session settings. These settings control the session configuration when there is no explicit policy overriding them.

To configure the session settings by using the command line interface

At the command prompt, type the following commands to configure the global session settings and verify the configuration:

“‘set tm sessionParameter [-sessTimeout][-defaultAuthorizationAction (ALLOW | DENY)][-SSO (ON | OFF)][-ssoCredential (PRIMARY | SECONDARY)][-ssoDomain][-httpOnlyCookie (YES | NO)][-persistentCookie (ENABLED | DISABLED)] [-persistentCookieValidity]

```
Example
```

> set tm sessionParameter -sessTimeout 30
> Done
> set tm sessionParameter -defaultAuthorizationAction DENY
> Done
> set tm sessionParameter -SSO ON
> Done
> set tm sessionParameter -ssoCredential PRIMARY
> Done

“‘

To configure the session settings by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic**
2. In the details pane, under **Settings**, click Change global settings.
3. In the **Global Session Settings** dialog, type or select values for the parameters.
    - Session Time-out—sessTimeout
    - Default Authorization Action—defaultAuthorizationAction
    - Single Sign-on to Web Applications—sso
    - Credential Index—ssoCredential
    - Single Sign-on Domain—ssoDomain
    - HTTPOnly Cookie—httpOnlyCookie

    - Enable Persistent Cookie—persistentCookie
    - Persistent Cookie Validity (minutes)—persistentCookieValidity
    - Home Page—homepage
4. Click **OK**.

Traffic Settings

September 21, 2020

If you use forms-based or SAML single sign-on (SSO) for your protected applications, you configure that feature in the Traffic settings. SSO enables your users to log on once to access all protected applications, rather than requiring them to log on separately to access each one.

Forms-based SSO allows you to use a web form of your own design as the sign-on method instead of a generic pop-up window. You can therefore put you company logo and other information you might want your users to see on the logon form. SAML SSO allows you to configure one NetScaler appliance or virtual appliance instance to authenticate to another NetScaler appliance on behalf of users who have authenticated with the first appliance.

To configure either type of SSO, you first create a forms or SAML SSO profile. Next, you create a traffic profile and link it to the SSO profile you created. Next, you create a policy, link it to the traffic profile. Finally, you bind the policy globally or to an authentication virtual server to put your configuration into effect.

Traffic profiles

September 21, 2020

After creating at least one forms or SAML sso profile, you must next create a traffic profile.

> **Note**
>
> In this feature, the terms "profile" and "action" mean the same thing.

To create a traffic profile by using the command line interface

At the command prompt, type:

"'add tm trafficAction [-appTimeout][-SSO (ON | OFF) [-formSSOAction]][-persistentCookie (ENABLED | DISABLED)][-InitiateLogout (ON | OFF)]

```
### Example
```

add tm trafficAction Traffic-Prof-1 –appTimeout 10 -SSO ON -formSSOAction SSO-Prof-1

~~~
## To modify a session profile by using the command line interface

At the command prompt, type:

```set tm trafficAction <name> [-appTimeout <mins>] [-SSO ( ON | OFF )
    [-formSSOAction <string>]] [-persistentCookie ( ENABLED | DISABLED )
    ] [-InitiateLogout ( ON | OFF )]<!--NeedCopy-->
~~~

Example

```
set tm trafficAction Traffic-Prof-1  - appTimeout 10 -SSO ON -
    formSSOAction SSO-Prof-1
<!--NeedCopy-->
```

To remove a session profile by using the command line interface

At the command prompt, type:

"'rm tm trafficAction

```
### Example
```

rm tm trafficAction Traffic-Prof-1

"'

To configure traffic profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Traffic.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**
3. In the details pane, click the Profiles tab.
4. On the Profiles tab, do one of the following:
    - To create a new traffic profile, click **Add.**
    - To modify an existing traffic profile, select the profile, and then click **Edit.**
5. In the **Create Traffic Profile** or **Configure Traffic Profile** dialog box, specify values for the parameters.
    - Name*—name (Cannot be changed for a previously configured session action.)

    - AppTimeout—appTimeout
    - Single Sign-On—SSO
    - Form SSO Action—formSSOAction
    - SAML SSO Action—samlSSOAction
    - Enable Persistent Cookie—persistentCookie
    - Initiate Logout—InitiateLogout
6. Click **Create** or **OK**. The traffic profile that you created appears in the Traffic Policies, Profiles, and either the Form SSO Profiles or SAML SSO Profiles pane, as appropriate.

Traffic policies

September 21, 2020

After you create one or more form SSO and traffic profiles, you create traffic policies and then bind the policies, either globally or to a traffic management virtual server, to put them into effect.

To create a traffic policy by using the command line interface

At the command prompt, type:

“‘add tm trafficPolicy

```
### Example
```

add tm trafficPolicy Traffic-Pol-1 “HTTP.REQ.HEADER(“Cookie”).CONTAINS(“login=true”)” Traffic-Prof-1

~~~
## To modify a traffic policy by using the command line interface

At the command prompt, type:

```set tm trafficPolicy <name> <rule> <action><!--NeedCopy-->
~~~

Example

```
set tm trafficPolicy Traffic-Pol-1 "HTTP.REQ.HEADER("Cookie").CONTAINS(
    "login=true")" Traffic-Prof-1
<!--NeedCopy-->
```

To globally bind a traffic policy by using the command line interface

At the command prompt, type:

"‘bind tm global -policyName [-priority]

~~~
### Example

```bind tm global -policyName Traffic-Pol-1<!--NeedCopy-->
~~~

To bind a traffic policy to a load balancing or content switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

- `bind lb vserver <name> -policy <policyName> [-priority <priority>]<!--NeedCopy-->`
- `bind cs vserver <name> -policy <policyName> [-priority <priority>]<!--NeedCopy-->`

Example

```
bind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1 -
    priority 1000
<!--NeedCopy-->
```

To unbind a globally bound traffic policy by using the command line interface

At the command prompt, type:

"‘unbind tm global -policyName

```
### Example
```

unbind tm global -policyName Traffic-Pol-1

```
## To unbind a traffic policy from a load balancing or content
    switching virtual server by using the command line interface

At the command prompt, type one of the following commands:

-  ```unbind lb vserver <name> -policy <policyname><!--NeedCopy-->
```

- `unbind cs vserver <name> -policy <policyname><!--NeedCopy-->`

Example

```
unbind authentication vserver auth-vserver-1 -policyName Traffic-Pol-1

<!--NeedCopy-->
```

To remove a traffic policy by using the command line interface

First unbind the session policy from global, and then, at the command prompt, type:

"'rm tm trafficPolicy

```
### Example
```

rm tm trafficPolicy Traffic-Pol-1
"'

To configure and bind traffic policies by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Traffic.**
2. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**
3. In the details pane, do one of the following:
    - To create a new session policy, click **Add.**
    - To modify an existing session policy, select the policy, and then click **Edit.**
4. In the **Create Traffic Policy** or **Configure Traffic Policy** dialog, specify values for the parameters.
    - Name*—policyName (Cannot be changed for a previously configured session policy.)
    - Profile*—actionName
    - Expression—rule (You enter expressions by first choosing the type of expression in the leftmost drop-down list beneath the Expression text area and then typing your expression directly into the expression text area, or by clicking Add to open the Add Expression dialog box and using the drop-down lists in it to construct your expression.)
5. Click **Create** or **OK**. The policy that you created appears in the details pane of the **Session Policies** and **Profiles** page.

Form SSO profiles

September 21, 2020

To enable and configure forms-based SSO, you first create an SSO profile.

> **Note**
>
> - Forms-based single sign-on does not work if the form is customized to include Javascript.
> - In this feature, the terms "profile" and "action" mean the same thing.

To create a form SSO profile by using the command line interface

At the command prompt, type:

- `add tm formSSOAction <name> -actionURL <URL> -userField <string> -passwdField <string> -ssoSuccessRule <expression> [-nameValuePair <string>] [-responsesize <positive_integer>][-nvtype ( STATIC | DYNAMIC )][-submitMethod ( GET | POST )]<!--NeedCopy-->`
- `show tm formSSOAction [<name>]<!--NeedCopy-->`

Example

```
add tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-nameValuePair "loginID passwd" -responsesize "9096"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nvtype STATIC -submitMethod GET
 – sessTimeout 10 -defaultAuthorizationAction ALLOW
<!--NeedCopy-->
```

To modify a form SSO by using the command line interface

At the command prompt, type:

"'set tm formSSOAction -actionURL -userField -passwdField -ssoSuccessRule [-nameValuePair] [-responsesize][-nvtype (STATIC | DYNAMIC)][-submitMethod (GET | POST)]

```
### Example
```

set tm formSSOAction SSO-Prof-1 -actionURL "/logon.php"
-userField "loginID" -passwdField "passwd"
-ssoSuccessRule "HTTP.RES.HEADER("Set-Cookie").CONTAINS("LogonID")"
-nameValuePair "loginID passwd" -responsesize "9096"
-nvtype STATIC -submitMethod GET
–sessTimeout 10 -defaultAuthorizationAction ALLOW

```
## To remove a form SSO profile by using the command line interface

```

~~~
At the command prompt, type:

```rm tm formSSOAction <name><!--NeedCopy-->
~~~

Example

```
rm tm sessionAction SSO-Prof-1
<!--NeedCopy-->
```

To configure form SSO profiles by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Traffic**.
2. In the details pane, click the **Form SSO Profiles** tab.
3. On the Form SSO Profiles tab, do one of the following:
    - To create a new form SSO profile, click **Add**.
    - To modify an existing form SSO profile, select the profile, and then click Edit.
4. In the **Create Form SSO Profile** or **Configure Form SSO Profile** dialog, specify values for the parameters:
    - Name*—name (Cannot be changed for a previously configured session action.)
    - Action URL*—actionURL
    - User Name Field*—userField
    - Password Field*—passField
    - Expression*—ssoSuccessRule
    - Name Value Pair—nameValuePair
    - Response Size—responsesize
    - Extraction—nvtype
    - Submit Method—submitMethod
5. Click **Create** or **OK**, and then click **Close**. The form SSO profile that you created appears in the **Traffic Policies**, **Profiles**, and **Form SSO Profiles** pane.

SAML SSO profiles

September 21, 2020

To enable and configure SAML-based SSO, you first create a SAML SSO profile.

To create a SAML SSO profile by using the command line interface

At the command prompt, type:

"'add tm samlSSOProfile -samlSigningCertName -assertionConsumerServiceURL -relaystateRule -sendPassword (ON | OFF) [-samlIssuerName]

```
### Example
```

add tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example, Inc." -assertionConsumerServiceURL "https://service.example.com" -relaystateRule "true" -sendPassword "ON" -samlIssuerName "Example, Inc."

~~~
## To modify a SAML SSO by using the command line interface

At the command prompt, type:

```set tm samlSSOProfile <name> -samlSigningCertName <string> -
    assertionConsumerServiceURL <URL> -relaystateRule <expression> -
    sendPassword (ON | OFF) [-samlIssuerName <string>]<!--NeedCopy-->
~~~

Example

```
set tm samlSSOProfile saml-SSO-Prof-1 -samlSigningCertName "Example,
    Inc."  -assertionConsumerServiceURL "https://service.example.com" -
    relaystateRule "true"  -sendPassword "ON" -samlIssuerName "Example,
    Inc."
<!--NeedCopy-->
```

To remove a SAML SSO profile by using the command line interface

At the command prompt, type:

"'rm tm samlSSOProfile

```
### Example
```

rm tm sessionAction saml-SSO-Prof-1

"'

To configure a SAML SSO profile by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Policies > Traffic.**

2. In the details pane, click the **SAML SSO Profiles** tab.
3. On the **SAML SSO Profiles** tab, do one of the following:
    - To create a new SAML SSO profile, click **Add.**
    - To modify an existing SAML SSO profile, select the profile, and then click **OpenEdit**.
4. In the **Create SAML SSO Profiles** or the **Configure SAML SSO Profiles** dialog box, set the following parameters:
    - Name*
    - Signing Certificate Name*
    - ACS URL*
    - Relay State Rule*
    - Send Password
    - Issuer Name
5. Click **Create** or **OK**, and then click **Close.** The SAML SSO profile that you created appears in the Traffic Policies, Profiles, and SAML SSO Profiles pane.

Session timeout for OWA 2010

September 21, 2020

You can now force OWA 2010 connections to time out after a specified period of inactivity. OWA sends repeated keepalive requests to the server to prevent timeouts. Keeping the connections open can interfere with single sign-on.

To force OWA 2010 to timeout after a specified period by using the command line interface

At the command prompt, type the following commands:

-
  ```
  add tm trafficAction <actname> [-forcedTimeout <forcedTimeout> -forcedTimeoutVal
   <mins>]<!--NeedCopy-->
  ```

  For <actname>, substitute a name for your traffic policy. For <mins>, substitute the number of minutes after which to initiate a forced timeout. For <forcedTimeout>, substitute one of the following values:

    - **START**—Starts the timer for forced timeout if a timer has not already been started. If a running timer exists, has no effect.
    - **STOP**—Stops a running timer. If no running timer is found, has no effect.
    - **RESET**—Restarts a running timer. If no running timer is found, starts a timer just as if the START option had been used.

- `add tm trafficPolicy <polname> <rule> <actname><!--NeedCopy-->`

  For <polname>, substitute a name for your traffic policy. For <rule>, substitute a rule in NetScaler appliance default syntax.

- `bind lb vserver <vservername> -policyName <name> -priority <number><!--NeedCopy-->`

  For <vservername>, substitute the name of the AAA traffic management virtual server. For <priority>, substitute an integer that designates the policy's priority.

Example

```
add tm trafficAction act-owa2010timeout -forcedTimeout RESET -
    forcedTimeoutVal 10
add tm trafficPolicy pol-owa2010timeout true act-owa2010timeout
bind lb vserver vs-owa2010 -policyName pol-owa2010timeout -priority 10
<!--NeedCopy-->
```

Authenticating with client certificates

September 21, 2020

Web sites that contain sensitive content, such as online banking websites or websites with employee personal information, sometimes require client certificates for authentication. To configure AAA to authenticate users on the basis of client-side certificate attributes, you first enable client authentication on the traffic management virtual server and bind the root certificate to the authentication virtual server. Then, you implement one of two options. You can configure the default authentication type on the authentication virtual server as CERT, or you can create a certificate action that defines what the NetScaler appliance must do to authenticate users on the basis of a client certificate. In either case, your authentication server must support CRLs. You configure the ADC to extract the user name from the SubjectCN field or another specified field in the client certificate.

When the user tries to log on to an authentication virtual server for which an authentication policy is not configured, and a global cascade is not configured, the user name information is extracted from the specified field of the certificate. If the required field is extracted, the authentication succeeds. If the user does not provide a valid certificate during the SSL handshake, or if the user name extraction fails, authentication fails. After it validates the client certificate, the ADC presents a logon page to the user.

The following procedures assume that you have already created a functioning AAA configuration, and therefore they explain only how to enable authentication by using client certificates. These procedures also assume that you have obtained your root certificate and client certificates and have placed them on the ADC in the /nsconfig/ssl directory.

To configure the AAA client certificate parameters by using the command line interface

At the command prompt, type the following commands, in the order shown, to configure the certificate and verify the configuration:

- `add ssl certKey <certkeyName> -cert <certFile> -key <keyFile> -password -inform <inform> -expiryMonitor <expiryMonitor> -notificationPeriod <notificationPeriod><!--NeedCopy-->`
- `bind ssl certKey <certkeyName> -vServer <certkeyName> -CA -crlCheck Mandatory<!--NeedCopy-->`
- `show ssl certKey [<certkeyName>]<!--NeedCopy-->`
- `set aaa parameter -defaultAuthType CERT<!--NeedCopy-->`
- `show aaa parameter<!--NeedCopy-->`
- `set aaa certParams -userNameField "Subject:CN"<!--NeedCopy-->`
- `show aaa certParams<!--NeedCopy-->`

To configure the AAA client certificate parameters by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic > Virtual Servers.**
2. In the details pane, select the virtual server that you want to configure to handle client certificate authentication, and then click **Edit.**
3. On the **Configuration** page, under **Certificates**, click the right arrow (>) to open the CA Cert Key installation dialog.
4. In the **CA Cert Key** dialog box, click **Insert.**
5. In the **CA Cert Key - SSL Certificates** dialog box, click **Install.**
6. In the **Install Certificate** dialog box, set the following parameters, whose names correspond to the CLI parameter names as shown:
   - Certificate-Key Pair Name*—certkeyName
   - Certificate File Name—certFile
   - Key File Name—keyFile
   - Certificate Format—inform
   - Password—password
   - Certificate Bundle—bundle
   - Notify When Expires—expiryMonitor
   - Notification Period—notificationPeriod

7. Click **Install**, and then click **Close**.
8. In the **CA Cert Key** dialog box, in the **Certificate** list, select the root certificate.
9. Click **Save**.
10. Click **Back** to return to the main configuration screen.
11. Navigate to **Security > AAA - Application Traffic > Policies > Authentication > CERT**.
12. In the details pane, select the policy you want to configure to handle client certificate authentication, and then click **Edit**.
13. In the **Configure Authentication CERT Policy** dialog, Server drop-down list, select the virtual server you just configured to handle client certificate authentication.
14. Click **OK**. A message appears in the status bar, stating that the configuration completed successfully.

Support to notify number of unsuccessful login attempts

The NetScaler appliance can now log the number of unsuccessful login attempts made from the last successful log on. The feature works only if the persistentLoginAttempts option is enabled on the appliance. By default, the option is disabled on the NetScaler appliance.

A NetScaler administrator can use this information to verify if any unauthorized attempts have occurred on a secured external user account.

To use this feature, at the NetScaler appliance command prompt, type:

“‘set aaa parameter [–maxloginAttempts [-failedLoginTimeout]] -persistentLoginAttempts (ENABLED | DISABLED)

~~~
**Example:**

```set aaa parameter  - maxLoginAttempts 4  - failedLoginTimeout 3  - persistentLoginAttempts ENABLED<!--NeedCopy-->
~~~

Client certificate pass-through

September 21, 2020

The NetScaler appliance can now be configured to pass client certificates through to protected applications that require client certificates for user authentication. The ADC first authenticates the user, then inserts the client certificate into the request and sends it to the application. This feature is configured by adding appropriate SSL policies.

The exact behavior of this feature when a user presents a client certificate depends upon the configuration of the VPN virtual server.

- If the VPN virtual server is configured to accept client certificates but not require them, the ADC inserts the certificate into the request and then forwards the request to the protected application.
- If the VPN virtual server has client certificate authentication disabled, the ADC renegotiatiates the authentication protocol and reauthenticates the user before it inserts the client certificate in the header and forwards the request to the protected application.
- If the VPN virtual server is configured to require client certificate authentication, the ADC uses the client certificate to authenticate the user, then inserts the certificate in the header and forwards the request to the protected application.

In all of these cases, you configure client certificate pass-through as follows.

To create and configure client certificate pass-through by using the command line interface

At the command prompt, type the following commands:

-
  ```
  add vpn vserver <name> SSL <IP> 443<!--NeedCopy-->
  ```

  For *name*, substitute a name for the virtual server. The name must contain from one to 127 ASCII characters, beginning with a letter or underscore (_), and containing only letters, numbers, and the underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. For <IP>, substitute the IP address assigned to the virtual server.“‘

-
  ```
  set ssl vserver <name> -clientAuth ENABLED -clientCert <clientcert>
  ```

  For <name>, substitute the name of the virtual server that you just created. For <clientCert>, substitute one of the following values:

  - disabled—disables client certificate authentication on the VPN virtual server.
  - mandatory—configures the VPN virtual server to require client certificates to authenticate.
  - optional—configures the VPN virtual server to allow client certificate authentication, but not to require it.

-
  ```
  bind vpn vserver \<name\> -policy local
  ```

  For <name>, substitute the name of the VPN virtual server that you created.

-
  ```
  bind vpn vserver \<name> -policy cert
  ```

  For <name>, substitute the name of the VPN virtual server that you created.

-
  ```
  bind ssl vserver \<name> -certkeyName \<certkeyname>
  ```

  For <name>, substitute the name of the virtual server that you created. For <certkeyName>, substitute the client certificate key.

-
  ```
  bind ssl vserver \<name> -certkeyName \<cacertkeyname> -CA -ocspCheck Optional
  ```

For <name>, substitute the name of the virtual server that you created. For <cacertkeyName>, substitute the CA certificate key.

- `add ssl action \<actname\> -clientCert ENABLED -certHeader CLIENT-CERT`
  For <actname>, substitute a name for the SSL action.
- `add ssl policy \<polname\> -rule true -action \<actname\>`
  For <polname>, substitute a name for your new SSL policy. For <actname>, substitute the name of the SSL action that you just created.
- `bind ssl vserver \<name\> -policyName \<polname\> -priority 10`
  For <name>, substitute the name of the VPN virtual server.

Example

‘‘‘

add vpn vserver vs-certpassthru SSL 10.121.250.75 443
set ssl vserver vs-certpassthru -clientAuth ENABLED -clientCert optional
bind vpn vserver vs-certpassthru -policy local
bind vpn vserver vs-certpassthru -policy cert
bind ssl vserver vs-certpassthru -certkeyName mycertKey
bind ssl vserver vs-certpassthru -certkeyName mycertKey -CA -ocspCheck Optional
add ssl action act-certpassthru -clientCert ENABLED -certHeader CLIENT-CERT
add ssl policy pol-certpassthru -rule true -action act-certpassthru
bind ssl vserver vs-certpassthru -policyName pol-certpassthru -priority 10

Configuring AAA with commonly used protocols

September 21, 2020

Configuring the NetScaler appliance for AAA needs a specific setup on the NetScaler appliance and clients’ browsers. The configuration varies with the protocol used for AAA.

For more information about configuring the NetScaler appliance for Kerberos authentication, see Handling Authentication, Authorization and Auditing with Kerberos/NTLM.

Handling authentication, authorization and auditing with Kerberos/NTLM

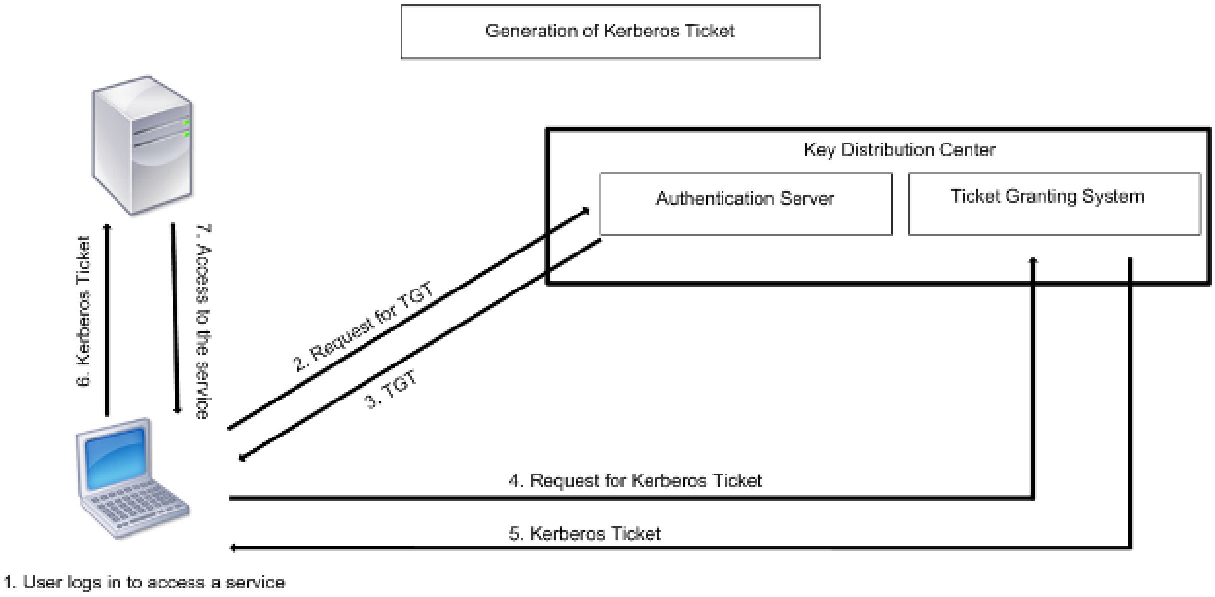
September 21, 2020

Kerberos, a computer network authentication protocol, provides secure communication over the Internet. Designed primarily for client-server applications, it provides for mutual authentication by which the client and server can each ensure the other's authenticity. Kerberos uses a trusted third party, referred to as Key Distribution Center (KDC). A KDC consists of an Authentication Server (AS), which authenticates a user, and a Ticket Granting Server (TGS).

Each entity on the network (client or server) has a secret key that is known only to itself and the KDC. The knowledge of this key implies authenticity of the entity. For communication between two entities on the network, the KDC generates a session key, referred to as the Kerberos ticket or service ticket. The client makes a request to the AS for credentials for a specific server. The client then receives a ticket, referred to as Ticket Granting Ticket (TGT). The client then contacts the TGS, using the TGT it received from the AS to prove its identity, and asks for a service. If the client is eligible for the service, the TGS issues a Kerberos ticket to the client. The client then contacts the server hosting the service (referred to as the service server), using the Kerberos ticket to prove that it is authorized to receive the service. The Kerberos ticket has a configurable lifetime. The client authenticates itself with the AS only once. If it contacts the physical server multiple times, it reuses the AS ticket.

The following figure shows the basic functioning of the Kerberos protocol.

Figure 1. **Functioning of Kerberos**



Kerberos authentication has the following advantages:

- Faster authentication. When a physical server gets a Kerberos ticket from a client, the server has enough information to authenticate the client directly. It does not have to contact a domain controller for client authentication, and therefore the authentication process is faster.
- Mutual authentication. When the KDC issues a Kerberos ticket to a client and the client uses the ticket to access a service, only authenticated servers can decrypt the Kerberos ticket. If the virtual server on the NetScaler appliance is able to decrypt the Kerberos ticket, you can

conclude that both the virtual server and client are authenticated. Thus, the authentication of the server happens along with the authentication of the client.
- Single sign-on between Windows and other operating systems that support Kerberos.

Kerberos authentication may have the following disadvantages:

- Kerberos has strict time requirements; the clocks of the involved hosts must be synchronized with the Kerberos server clock to ensure that the authentication does not fail. You can mitigate this disadvantage by using the Network Time Protocol daemons to keep the host clocks synchronized. Kerberos tickets have an availability period, which you can configure.
- Kerberos needs the central server to be available continuously. When the Kerberos server is down, no one can log on. You can mitigate this risk by using multiple Kerberos servers and fallback authentication mechanisms.
- Because all the authentication is controlled by a centralized KDC, any compromise in this infrastructure, such as the user's password for a local workstation being stolen, can allow an attacker to impersonate any user. You can mitigate this risk to some extent by using only a desktop machine or laptop that you trust, or by enforcing preauthentication by means of a hardware-token.

To use Kerberos authentication, you must configure it on the NetScaler appliance and on each client.

Optimizing Kerberos authentication on AAA

The NetScaler appliance now optimizes and improves the system performance while Kerberos authentication. The AAA daemon remembers the outstanding Kerberos request for the same user to avoid load on Key Distribution Center (KDC), which will avoid duplicate requests.

How NetScaler appliance implements Kerberos for client authentication

September 21, 2020

> **Important**
>
> Kerberos/NTLM authentication is supported only in the NetScaler 9.3 nCore release or later, and it can be used only for AAA traffic management virtual servers.

NetScaler appliance handles the components involved in Kerberos authentication in the following way:

Key Distribution Center (KDC)

In the Windows 2000 Server or later versions, the Domain Controller and KDC are part of the Windows Server. If the Windows Server is UP and running, it indicates that the Domain Controller and KDC are configured. The KDC is also the Active Directory server.

> **Note**
>
> All Kerberos interactions are validated with the Windows Kerberos Domain Controller.

Authentication service and protocol negotiation

A NetScaler appliance supports Kerberos authentication on the AAA traffic management authentication virtual servers. If the Kerberos authentication fails, the NetScaler appliance uses the NTLM authentication.

By default, Windows 2000 Server and later Windows Server versions use Kerberos for AAA. If you create an authentication policy with NEGOTIATE as the authentication type, the NetScaler appliance attempts to use the Kerberos protocol for AAA and if the client's browser fails to receive a Kerberos ticket, the NetScaler appliance uses the NTLM authentication. This process is referred to as negotiation.

The client may fail to receive a Kerberos ticket in any of the following cases:

- Kerberos is not supported on the client.
- Kerberos is not enabled on the client.
- The client is in a domain other than that of the KDC.
- The Access Directory on the KDC is not accessible to the client.

For Kerberos/NTLM authentication, the NetScaler appliance does not use the data that is present locally on the NetScaler appliance.

Authorization

The traffic management virtual server can be a load balancing virtual server or a content switching virtual server.

Auditing

The NetScaler appliance supports auditing of Kerberos authentication with the following audit logging:

- Complete audit trail of the traffic management end-user activity
- SYSLOG and high performance TCP logging
- Complete audit trail of system administrators
- All system events

- Scriptable log format

Supported Environment

Kerberos authentication does not need any specific environment on the NetScaler appliance. The client (browser) must provide support for Kerberos authentication.

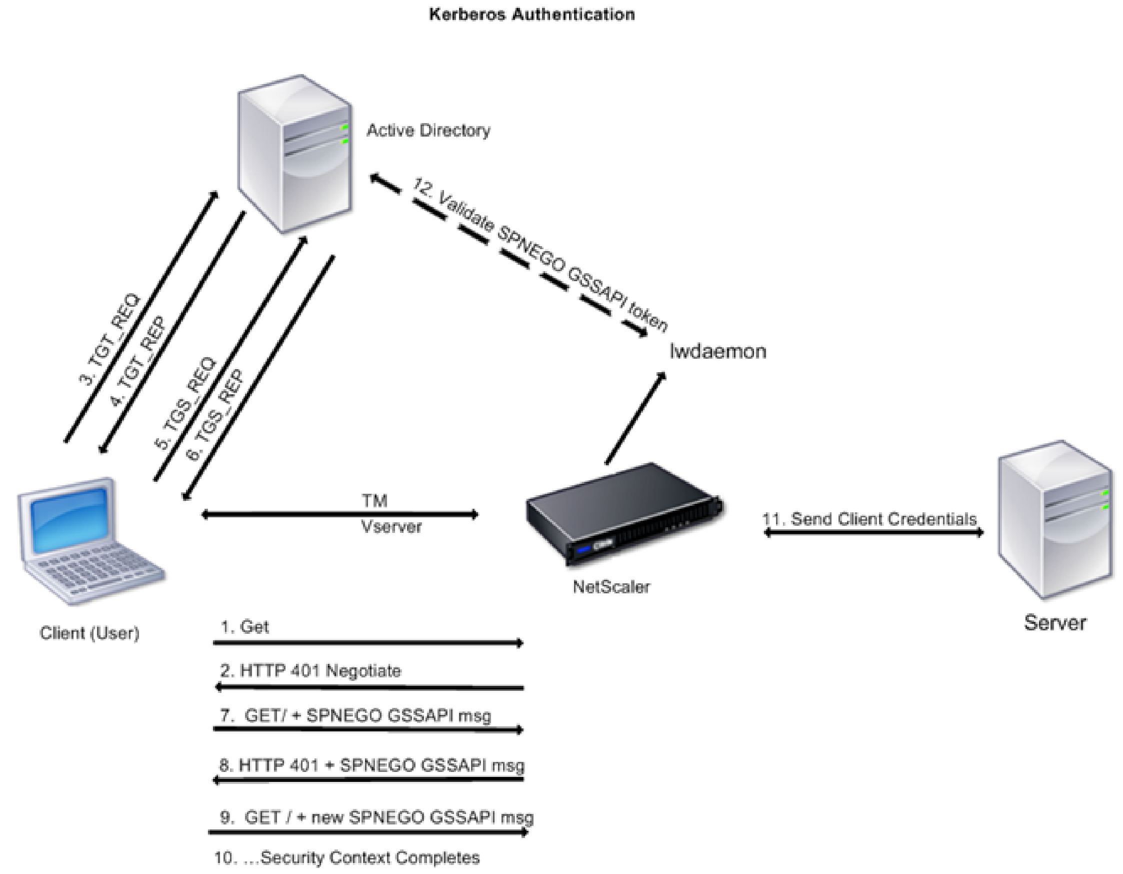
High Availability

In a high availability setup, only the active NetScaler appliance joins the domain. In case of a failover, the NetScaler appliance lwagent daemon joins the secondary NetScaler appliance to the domain. No specific configuration is required for this functionality.

Kerberos authentication process

The following figure shows a typical process for Kerberos authentication in the NetScaler appliance environment.

Figure 1. Kerberos Authentication Process on NetScaler appliance


Kerberos Authentication
Active Directory
12. Validate SPNEGO GSSAPI token
lwdaemon
3. TGT_REQ
4. TGT_REP
5. TGS_REQ
6. TGS_REP
TM
Vserver
NetScaler
11. Send Client Credentials
Server
Client (User)
1. Get
2. HTTP 401 Negotiate
7. GET/ + SPNEGO GSSAPI msg
8. HTTP 401 + SPNEGO GSSAPI msg
9. GET / + new SPNEGO GSSAPI msg
10. …Security Context Completes

The Kerberos authentication occurs in the following stages:

Client authenticates itself to the KDC

1. The NetScaler appliance receives a request from a client.
2. The traffic management (load balancing or content switching) virtual server on the NetScaler appliance sends a challenge to the client.
3. To respond to the challenge, the client gets a Kerberos ticket.
    - The client sends the Authentication Server of the KDC a request for a ticket-granting ticket (TGT) and receives the TGT. (See 3, 4 in the figure, Kerberos Authentication Process.)
    - The client sends the TGT to the Ticket Granting Server of the KDC and receives a Kerberos ticket. (See 5, 6 in the figure, Kerberos Authentication Process.)

> Note
>
> The above authentication process is not necessary if the client already has a Kerberos ticket whose lifetime has not expired. In addition, clients such as Web Services, .NET, or J2EE, which support SPNEGO, get a Kerberos ticket for the target server, create an SPNEGO token, and insert the token in the HTTP header when they send an HTTP request. They do not go through the client authentication process.

Client requests a service.

1. The client sends the Kerberos ticket containing the SPNEGO token and the HTTP request to the traffic management virtual server on the NetScaler appliance. The SPNEGO token has the necessary GSSAPI data.
2. The NetScaler appliance establishes a security context between the client and the NetScaler appliance. If the NetScaler appliance cannot accept the data provided in the Kerberos ticket, the client is asked to get a different ticket. This cycle repeats till the GSSAPI data is acceptable and the security context is established. The traffic management virtual server on the NetScaler appliance acts as an HTTP proxy between the client and the physical server.

NetScaler appliance completes the authentication.

1. After the security context is complete, the traffic management virtual server validates the SPNEGO token.
2. From the valid SPNEGO token, the virtual server extracts the user ID and GSS credentials, and passes them to the authentication daemon.
3. A successful authentication completes the Kerberos authentication.

Configuring kerberos authentication on the NetScaler appliance

September 21, 2020

This topic provides the detailed steps to configure Kerberos authentication on the NetScaler appliance by using the CLI and the GUI.

Configuring Kerberos authentication on the CLI

1. Enable the AAA feature to ensure the authentication of traffic on the appliance.

   ```
   ns-cli-prompt> enable ns feature AAA
   ```

2. Add the keytab file to the NetScaler appliance. A keytab file is necessary for decrypting the secret received from the client during Kerberos authentication. A single keytab file contains authentication details for all the services that are bound to the traffic management virtual server on the NetScaler appliance.

   First generate the keytab file on the Active Directory server and then transfer it to the NetScaler appliance.

   a) Log on to the Active Directory server and add a user for Kerberos authentication. For example, to add a user named “Kerb-SVC-Account”:

   ```
   net user Kerb-SVC-Account freebsd!@#456 /add
   ```

   > **Note**
   >
   > In the **User Properties** section, ensure that the “Change password at next logon option” is not selected and the “Password does not expire” option is selected.

   b) Map the HTTP service to the above user and export the keytab file. For example, run the following command on the Active Directory server:

   ```
   ktpass /out keytabfile /princ HTTP/owa.newacp.com@NEWACP.COM /pass freebsd!@#456
   /mapuser newacp\dummy /ptype KRB5_NT_PRINCIPAL
   ```

   > **Note**
   >
   > You can map more than one service if authentication is required for more than one service. If you want to map more services, repeat the above command for every service. You can give the same name or different names for the output file.

   c) Transfer the keytab file to the NetScaler appliance by using the unix **ftp** command or any other file transfer utility of your choice.

   d) Log on to the NetScaler appliance, and run the **ktutil** utility to verify the keytab file. The keytab file has an entry for the HTTP service after it is imported.

   The **kutil** interactions are as follows:

   ```
   root@ns# ktutil
   ktutil: rkt /var/keytabfile
   ktutil: list
   ```

```
slot KVNO Principal
——————————————————————————

ktutil: wkt /etc/ krb5.keytab
ktutil: list

slot KVNO Principal
—- —- ————————————————————————-

1 2 HTTP/owa.newacp.com@NEWACP.COM

ktutil: quit
```

3. The NetScaler appliance must obtain the IP address of the domain controller from the fully qualified domain name (FQDN). Therefore, Citrix recommends configuring the NetScaler appliance with a DNS server.

   ```
   ns-cli-prompt> add dns nameserver <ip-address>
   ```

   > **Note**
   >
   > Alternatively, you can add static host entries or use any other means so that the NetScaler appliance can resolve the FQDN name of the domain controller to an IP address.

4. Configure the authentication action and then associate it to an authentication policy.

   a) Configure the negotiate action.

   ```
   ns-cli-prompt> add authentication negotiateAction <name> -domain <domainName> -domainUser <domainUsername> -domainUserPasswd <domainUserPassword>
   ```

   b) Configure the negotiate policy and associate the negotiate action to this policy.

   ```
   ns-cli-prompt> add authentication negotiatePolicy <name> <rule> <reqAction>
   ```

5. Create an authentication virtual server and associate the negotiate policy with it.

   a) Create an authentication virtual server.

   ```
   ns-cli-prompt> add authentication vserver <name> SSL <ipAuthVserver> 443 -authenticationDomain <domainName>
   ```

   b) Bind the negotiate policy to the authentication virtual server.

   ```
   ns-cli-prompt> bind authentication vserver <name> -policy <negotiatePolicyName>
   ```

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   ```
   ns-cli-prompt> set lb vserver <name> -authn401 ON -authnVsName <string>
   ```

> **Note**
>
> Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations by doing the following:
    a) Access the traffic management virtual server, using the FQDN. For example, Sample
    b) View the details of the session on the CLI.

       ns-cli-prompt> **show aaa session**

Configuring Kerberos authentication on the GUI

1. Enable the AAA feature.

   Navigate to **System** > **Settings**, click **Configure Basic Features** and enable the AAA feature.

2. Add the keytab file as detailed in step 2 of the CLI procedure mentioned above.
3. Add a DNS server.

   Navigate to **Traffic Management > DNS > Name Servers**, and specify the IP address for the DNS server.

4. Configure the **Negotiate** action and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with **Negotiate** as the action type.

5. Bind the negotiate policy to the authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the **Negotiate** policy with the authentication virtual server.

6. Associate the authentication virtual server with the traffic management (load balancing or content switching) virtual server.

   Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and specify the relevant authentication settings.

   > **Note**
   >
   > Similar configurations can also be done on the content switching virtual server.

7. Verify the configurations as detailed in step 7 of the CLI procedure mentioned above.

Configuring kerberos authentication on a client

September 21, 2020

Kerberos support must be configured on the browser to use Kerberos for authentication. You can use any Kerberos-compliant browser. Instructions for configuring Kerberos support on Internet Explorer and Mozilla Firefox follow. For other browsers, see the documentation of the browser.

To configure Internet Explorer for Kerberos authentication

1. In the **Tools** menu select **Internet Options**.
2. On the **Security** tab, click **Local Intranet**, and then click **Sites**.
3. In the **Local Intranet** dialog box, make sure that the Automatically detect intranet network option is selected, and then click **Advanced**.
4. In the **Local Intranet** dialog box, add the web sites of the domains of the traffic management virtual server on the NetScaler appliance. The specified sites become local intranet sites.
5. Click **Close** or **OK** to close the dialog boxes.

To configure Mozilla Firefox for Kerberos authentication

1. Make sure that you have Kerberos properly configured on your computer.
2. Type about:config in the URL bar.
3. In the filter text box, type network.negotiate.
4. Change network.negotiate-auth.delegation-uris to the domain that you want to add.
5. Change network.negotiate-auth.trusted-uris to the domain that you want to add.

   > Note: If you are running Windows, you also need to enter sspi in the filter text box and change the network.auth.use-sspi option to False.

Offloading Kerberos authentication from physical servers

September 21, 2020

The NetScaler appliance can offload authentication tasks from servers. Instead of the physical servers authenticating the requests from clients, the NetScaler appliance authenticates all the client requests before it forwards them to any of the physical servers bound to it. The user authentication is based on Active Directory tokens.

There is no authentication between the NetScaler appliance and the physical server, and the authentication offload is transparent to the end users. After the initial logon to a Windows computer, the end user does not have to enter any additional authentication information in a pop-up or on a logon page.

In the current NetScaler appliance release, Kerberos authentication is available only for AAA traffic management virtual servers. Kerberos authentication is not supported for SSL VPN in the Citrix Gateway Enterprise Edition appliance or for NetScaler appliance management.

Kerberos authentication requires configuration on the NetScaler appliance and on client browsers.

To configure Kerberos authentication on the NetScaler appliance

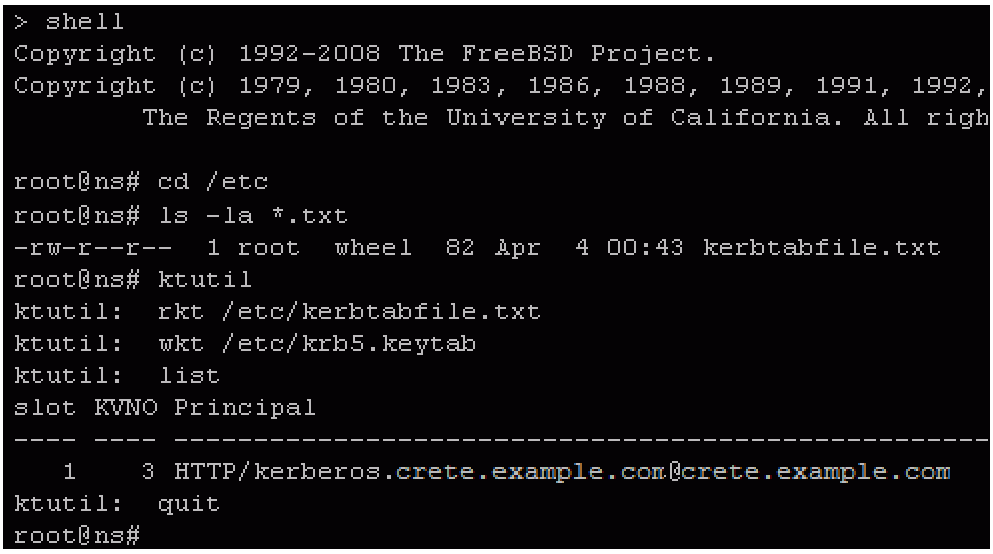
1. Create a user account on Active Directory. When creating a user account, verify the following options in the User Properties section:
   - Make sure that you do not select the Change password at next logon option.
   - Be sure to select the Password does not expire option.
2. On the AD server, at the CLI command prompt, type:
   - `ktpass -princ HTTP/kerberos.crete.lab.net@crete.lab.net -ptype KRB5_NT_PRINCIPAL -mapuser kerbuser@crete.lab.net -mapop set -pass Citrix1 -out C:\kerbtabfile.txt`

   > **Note**
   >
   > Be sure to type the above command on a single line. The output of the above command is written into the C:\kerbtabfile.txt file.
3. Upload the kerbtabfile.txt file to the /etc directory of the NetScaler appliance by using a Secure Copy (SCP) client.
4. Run the following command to add a DNS server to the NetScaler appliance.
   - `add dns nameserver 1.2.3.4`

   The NetScaler appliance cannot process Kerberos requests without the DNS server. Be sure to use the same DNS server that is used in the Microsoft Windows domain.
5. Switch to the shell prompt and run the following commands from the shell prompt:
   - `ktutil # rkt /etc/kerbtabfile.txt`
   - `# wkt /etc/krb5.keytab`
   - `# list`

     The list command displays the user account details that you created in the Active Directory. A sample screen of the output of the list command is shown below.

     Figure 1. Sample Output of the list Command



```
> shell
Copyright (c) 1992-2008 The FreeBSD Project.
Copyright (c) 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992,
        The Regents of the University of California. All righ

root@ns# cd /etc
root@ns# ls -la *.txt
-rw-r--r--  1 root  wheel  82 Apr  4 00:43 kerbtabfile.txt
root@ns# ktutil
ktutil:  rkt /etc/kerbtabfile.txt
ktutil:  wkt /etc/krb5.keytab
ktutil:  list
slot KVNO Principal
---- ---- ----------------------------------------------------
   1    3 HTTP/kerberos.crete.example.com@crete.example.com
ktutil:  quit
root@ns#
```

6. Switch to the command line interface of NetScaler appliance.
7. Run the following command to create a Kerberos authentication server:
    - add authentication negotiateAction KerberosServer -domain “crete.lab.net” -domainUser kerbuser -domainUserPasswd Citrix1
8. Run the following command to create a negotiation policy:
    - `add authentication negotiatePolicy Kerberos-Policy "REQ.IP.DESTIP == 192.168.17.200"KerberosServer<!--NeedCopy-->`
9. Run the following command to create an authentication virtual server.
    - `add authentication vserver Kerb-Auth SSL 192.168.17.201 443 -AuthenticationDomain crete.example.com<!--NeedCopy-->`
10. Run the following command to bind the Kerberos policy to the authentication virtual server:
    - `bind authentication vserver Kerb-Auth -policy Kerberos-Policy -priority 100<!--NeedCopy-->`
11. Run the following command to bind an SSL certificate to the authentication virtual server. You can use one of the test certificates, which you can install from the GUI NetScaler appliance. Run the following command to use the ServerTestCert sample certificate.
    - `bind ssl vserver Kerb-Auth -certkeyName ServerTestCert<!--NeedCopy-->`
12. Create an HTTP load balancing virtual server with the IP address, 192.168.17.200.

    Ensure that you create a virtual server from the command line interface for NetScaler 9.3 releases if they are older than 9.3.47.8.
13. Run the following command to configure an authentication virtual server:

- `set lb vserver <name>-authn401 ON -authnVsName Kerb-Auth<!--NeedCopy -->`

14. Enter the host name Example in the address bar of the Web browser.

    The Web browser displays an authentication dialog box because the Kerberos authentication is not set up in the browser.

    > **Note**
    >
    > Kerberos authentication requires a specific configuration on the client. Ensure that the client can resolve the hostname, which results in the Web browser connecting to an HTTP virtual server.

15. Configure Kerberos on the Web browser of the client computer.

    - For configuring on Internet Explorer, see Configuring Internet Explorer for Kerberos authentication.
    - For configuring on Mozilla Firefox, see Configuring Internet Explorer for Kerberos authentication.

16. Verify whether you can access the backend physical server without authentication.

To configure Internet Explorer for Kerberos authentication

1. Select **Internet Options** from the **Tools** menu.
2. Activate the **Security** tab.
3. Select **Local Intranet** from the Select a zone to view change security settings section.
4. Click **Sites**.
5. Click **Advanced**.
6. Specify the URL, Example and click **Add**.
7. Restart **Internet Explorer**.

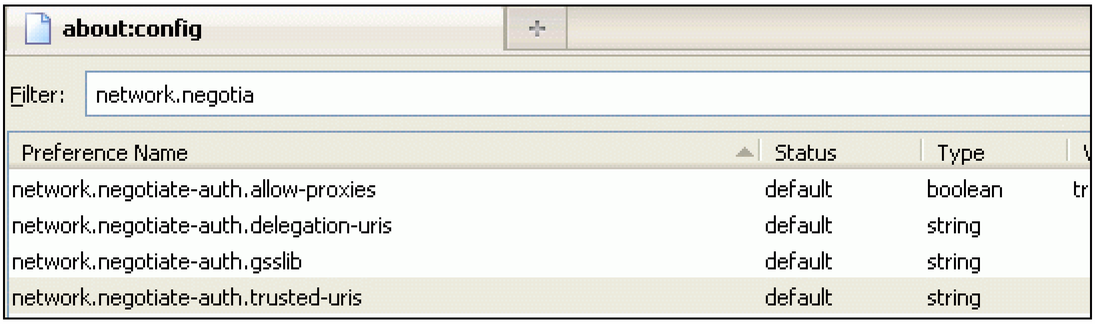
To configure Mozilla Firefox for Kerberos authentication

1. Enter about:config in the address bar of the browser.

2. Click the warning disclaimer.

3. Type **Network.Negotiate-auth.trusted-uris** in the **Filter** box.

4. Double click **Network.Negotiate-auth.trusted-uris.** A sample screen is shown below.



Filter: network.negotia

Preference Name	Status	Type
network.negotiate-auth.allow-proxies	default	boolean
network.negotiate-auth.delegation-uris	default	string
network.negotiate-auth.gsslib	default	string
network.negotiate-auth.trusted-uris	default	string

5. In the Enter String Value dialog box, specify www.crete.example.com.
6. Restart Firefox.

NetScaler appliance kerberos single sign-on

September 21, 2020

NetScaler appliances now support single sign-on (SSO) using the Kerberos 5 protocol. Users log on to a proxy, the Application Delivery Controller (ADC), which then provides access to protected resources.

The NetScaler appliance Kerberos SSO implementation requires the user's password for SSO methods that rely on basic, NTLM, or forms-based authentication. The user's password is not required for Kerberos SSO, although if Kerberos SSO fails and the NetScaler appliance has the user's password, it uses the password to attempt NTLM SSO.

If the user's password is available, the KCD account is configured with a realm, and no delegated user information is present, the NetScaler AD Kerberos SSO engine impersonates the user to obtain access to authorized resources. Impersonation is also called unconstrained delegation.

The NetScaler appliance Kerberos SSO engine can also be configured to use a delegated account to obtain access to protected resources on the user's behalf. This configuration requires delegated user credentials, a keytab, or a delegated user certificate and matching CA certificate. Configuration that uses a delegated account is called constrained delegation.

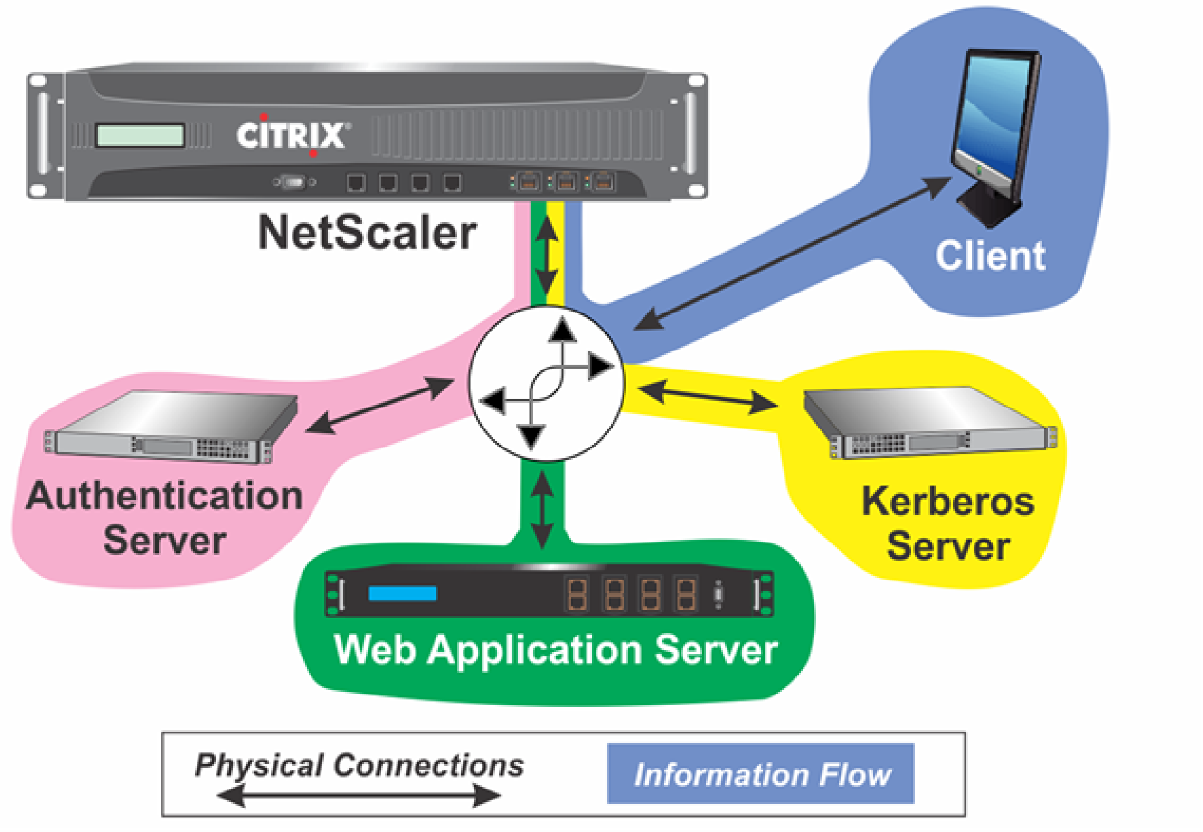
An overview of NetScaler appliance kerberos SSO

September 21, 2020

To use the NetScaler appliance Kerberos SSO feature, users first authenticate with Kerberos or a supported third-party authentication server. Once authenticated, the user requests access to a protected web application. The web server responds with a request for proof that the user is authorized to access that web application. The user's browser contacts the Kerberos server, which verifies that the user is authorized to access that resource, and then provides the user's browser with a service ticket that provides proof. The browser resends the user's request to the web application server with the service ticket attached. The web application server verifies the service ticket, and then allows the user to access the application.

AAA traffic management implements this process as shown in the following diagram. The diagram illustrates the flow of information through the NetScaler appliance and AAA traffic management, on a secure network with LDAP authentication and Kerberos authorization. AAA traffic management environments that use other types of authentication have essentially the same information flow, although they might differ in some details.

Figure 1. A Secure Network with LDAP and Kerberos



The AAA traffic management with authentication and authorization in a Kerberos environment requires that the following actions take place.

1. The client sends a request for a resource to the traffic management virtual server on the NetScaler appliance.
2. The traffic management virtual server passes the request to the authentication virtual server, which authenticates the client and then passes the request back to the traffic management vir-

tual server.

3. The traffic management virtual server sends the client's request to the web application server.
4. The web application server responds to the traffic management virtual server with a 401 Unauthorized message that requests Kerberos authentication, with fallback to NTLM authentication if the client does not support Kerberos.
5. The traffic management virtual server contacts the Kerberos SSO daemon.
6. The Kerberos SSO daemon contacts the Kerberos server and obtains a ticket-granting ticket (TGT) allowing it to request service tickets authorizing access to protected applications.
7. The Kerberos SSO daemon obtains a service ticket for the user and sends that ticket to the traffic management virtual server.
8. The traffic management virtual server attaches the ticket to the user's initial request and sends the modified request back to the web application server.
9. The web application server responds with a 200 OK message.

These steps are transparent to the client, which just sends a request and receives the requested resource.

Integration of NetScaler appliance Kerberos SSO with authentication methods

All AAA traffic management authentication mechanisms support NetScaler appliance Kerberos SSO. AAA traffic management supports the Kerberos SSO mechanism with the Kerberos, CAC (Smart Card) and SAML authentication mechanisms with any form of client authentication to the NetScaler appliance. It also supports the HTTP-Basic, HTTP-Digest, Forms-based, and NTLM (versions 1 and 2) SSO mechanisms if the client uses either HTTP-Basic or Forms-Based authentication to log on to the NetScaler appliance.

The following table shows each supported client-side authentication method, and the supported server-side authentication method for that client-side method.

Table 1. Supported Authentication Methods

	Basic/Digest/NTLM	Kerberos Constrained Delegation	User Impersonation
CAC (Smart Card): at SSL/T LS Layer		X	X
Forms-Based (LDAP/RADIUS/TACACS)	X	X	X
HTTP Basic (LDAP/RADIUS/TACACS)	X	X	X
Kerberos		X	

	Basic/Digest/NTLM	Kerberos Constrained Delegation	User Impersonation
NT LM v1/v2		X	X
SAML		X	
SAML Two-Factor	X	X	X
Certificate Two-Factor	X	X	X

Setting up NetScaler appliance SSO

September 21, 2020

You can configure NetScaler appliance SSO to work in one of two ways: by impersonation or by delegation. SSO by impersonation is a simpler configuration than SSO by delegation, and is therefore usually preferable when your configuration allows it. To configure NetScaler appliance SSO by impersonation, you must have the user's user name and password.

To configure NetScaler appliance SSO by delegation, you must have the delegated user's credentials in one of the following formats: the user's user name and password, the keytab configuration that includes the user name and an encrypted password, or the delegated user certificate and the matching CA certificate.

Prerequisites

September 21, 2020

Before you configure NetScaler appliance SSO, you need to have your NetScaler appliance fully configured to manage traffic to and authentication for your web application servers. Therefore, you must configure either load balancing or content switching, and then AAA, for these web application servers. You should also verify routing between the appliance, your LDAP server, and your Kerberos server.

If your network is not already configured in this manner, perform the following configuration tasks:

- Configure a server and service for each web application server.
- Configure a traffic management virtual server to handle traffic to and from your web application server.

Following are brief instructions and examples for performing each of these tasks from the NetScaler appliance command line. For further assistance, see Setting up an Authentication Virtual Server.

To create a server and service by using the command line

For NetScaler appliance SSO to obtain a TGS (service ticket) for a service, either the FQDN assigned to the server entity on the NetScaler appliance must match the FQDN of the web application server, or the server entity name must match the NetBios name of the web application server. You can take either of the following approaches:

- Configure the NetScaler appliance server entity by specifying the FQDN of the web application server.
- Configure the NetScaler appliance server entity by specifying the IP address of the web application server, and assign the server entity the same name as the NetBios name of the web application server.

At the command prompt, type the following commands:

```
-   add server name <serverFQDN>

-   add service name serverName serviceType port
<!--NeedCopy-->
```

For the variables, substitute the following values:

- **serverName**. A name for the NetScaler appliance to use to refer to this server.
- **serverFQDN**. The FQDN of the server. If the server has no domain assigned to it, use the server's IP address and make sure that the server entity name matches the NetBios name of the web application server.
- **serviceName**. A name for the NetScaler appliance to use to refer to this service.
- **type**. The protocol used by the service, either HTTP or MSSQLSVC.
- **port**. The port on which the service listens. HTTP services normally listen on port 80. Secure HTTPS services normally listen on port 443.

Example

The following examples add server and service entries on the NetScaler appliance for the web application server was1.example.com. The first example uses the FQDN of the web application server; the second uses the IP address.

To add the server and service using the web application server FQDN, was1.example.com, you would type the following commands:

```
add server was1 was1.example.com
add service was1service was1 HTTP 80
<!--NeedCopy-->
```

To add the server and service using the web application server IP and NetBios name, where the web application server IP is 10.237.64.87 and its NetBios name is WAS1, you would type the following commands:

```
add server WAS1 10.237.64.87
add service was1service WAS1 HTTP 8
<!--NeedCopy-->
```

To create a traffic management virtual server by using the command line

The traffic management virtual server manages traffic between the client and the web application server. You can use either a load balancing or a content switching virtual server as the traffic management server. The SSO configuration is the same for either type.

To create a load balancing virtual server, at the command prompt, type the following command:

"'add lb vserver

```
For the variables, substitute the following values:

-  **vserverName** – A name for the NetScaler appliance to use to refer
    to this virtual server.
-  **type** – The protocol used by the service, either HTTP or MSSQLSVC.
-  **IP** – The IP address assigned to the virtual server. This would
    normally be an IANA-reserved, non-public IP address on your LAN.
-  **port** – The port on which the service listens. HTTP services
    normally listen on port 80. Secure HTTPS services normally listen on
     port 443.

## Example

To add a load balancing virtual server called tmvserver1 to a
    configuration that manages HTTP traffic on port 80, assigning it a
    LAN IP address of 10.217.28.20 and then binding the load balancing
    virtual server to the wasservice1 service, you would type the
    following commands:
```

add lb vserver tmvserver1 HTTP 10.217.28.20 80
bind lb vserver tmvserv1 wasservice1

```
## To create an authentication virtual server by using the command line

The authentication virtual server manages authentication traffic
    between the client and the authentication (LDAP) server. To create
```

```
an authentication virtual server, at the command prompt type the following commands:
```

- add authentication vserver SSL 443
- set authentication vserver –authenticationdomain

```
For the variables, substitute the following values:

-  **authvserverName** — A name for the NetScaler appliance to use to refer to this authentication virtual server. Must begin with a letter, number, or the underscore character (\_), and must contain only letters, numbers, and the hyphen (-), period (.) pound (\#), space ( ), at (@), equals (=), colon (:), and underscore characters. Can be changed after the authentication virtual server is added by using the rename authentication vserver command.
-  **IP** — The IP address assigned to the authentication virtual server. As with the traffic management virtual server, this address would normally be an IANA-reserved, non-public IP on your LAN.
-  **domain** — The domain assigned to the virtual server. This would usually be the domain of your network. It is customary, though not required, to enter the domain in all capitals when configuring the authentication virtual server.

## Example

To add an authentication virtual server called authverver1 to your configuration and assign it the LAN IP 10.217.28.21 and the domain EXAMPLE.COM, you would type the following commands:
```

add authentication vserver authvserver1 SSL 10.217.28.21 443
set authentication vserver authvserver1 –authenticationdomain EXAMPLE.COM

```
## To configure a traffic management virtual server to use an authentication profile

The authentication virtual server can be configured to handle authentication for a single domain or for multiple domains. If it is configured to support authentication for multiple domains, you must also specify the domain for NetScaler appliance SSO by creating an authentication profile, and then configuring the traffic management virtual server to use that authentication profile.

> **Note**
>
> The traffic management virtual server can be either a load balancing
```

```
(lb) or content switching (cs) virtual server. The following instructions assume that you are using a load balancing virtual server. To configure a content switching virtual server, simply substitute
set cs vserver for set lb vserver. The procedure is otherwise the same.

To create the authentication profile, and then configure the authentication profile on a traffic management virtual server, type the following commands:
```

- add authentication authnProfile {-authvserverName } {-authenticationHost } {-authenticationDomain }
- set lb vserver <vserverName> -authnProfile

```
For the variables, substitute the following values:

- **authnprofileName** – A name for the authentication profile. Must begin with a letter, number, or the underscore character (\_), and must consist of from one to thirty-one alphanumeric or hyphen (-), period (.) pound (\#), space ( ), at (@), equals (=), colon (:), and underscore characters.
- **authvserverName** – The name of the authentication virtual server that this profile uses for authentication.
- **authenticationHost** – Host name of the authentication virtual server.
- **authenticationDomain** – Domain for which NetScaler appliance SSO handles authentication. Required if the authentication virtual server performs authentication for more than one domain, so that the correct domain is included when the NetScaler appliance sets the traffic management virtual server cookie.

## Example

To create an authentication profile named authnProfile1 for authentication of the example.com domain, and to configure the load balancing virtual server vserver1 to use the authentication profile authnProfile1, you would type the following commands:
```

add authentication authnProfile authnProfile1 -authnvsName authvsesrver1
-authenticationHost authvsesrver1 -authenticationDomain example.com
set lb vserver vserver1 -authnProfile authnProfile1

‘‘‘

Configuring SSO

September 21, 2020

Configuring NetScaler appliance SSO to authenticate by impersonation is simpler than configuring than SSO to authenticate by delegation, and is therefore usually preferable when your configuration allows it. You just create a KCD account. You can use the user's password.

If you do not have the user's password, you can configure NetScaler appliance SSO to authenticate by delegation. Although somewhat more complex than configuring SSO to authenticate by impersonation, the delegation method provides flexibility in that a user's credentials might not be available to the NetScaler appliance in all circumstances.

For either impersonation or delegation, you must also enable integrated authentication on the web application server.

Enabling integrated authentication on the web application server

September 21, 2020

To set up NetScaler appliance Kerberos SSO on each web application server that Kerberos SSO will manage, use the configuration interface on that server to configure the server to require authentication. Select Kerberos (negotiate) authentication by preference, with fallback to NTLM for clients that do not support Kerberos.

Following are instructions for configuring Microsoft Internet Information Server (IIS) to require authentication. If your web application server uses software other than IIS, consult the documentation for that web server software for instructions.

To configure Microsoft IIS to use integrated authentication

1. Log on to the IIS server and open **Internet Information Services Manager.**
2. Select the web site for which you want to enable integrated authentication. To enable integrated authentication for all IIS web servers managed by IISM, configure authentication settings for the Default Web Site. To enable integrated authentication for individual services (such as Exchange, Exadmin, ExchWeb, and Public), configure these authentication settings for each service individually.
3. Open the **Properties** dialog box for the default web site or for the individual service, and click the **Directory Security** tab.
4. Beside **Authentication** and **Access Control**, select **Edit.**
5. Disable anonymous access.

6. Enable Integrated Windows authentication (only). Enabling integrated Windows authentication should automatically set protocol negotiation for the web server to Negotiate, NTLM, which specifies Kerberos authentication with fallback to NTLM for non-Kerberos capable devices. If this option is not automatically selected, manually set protocol negotiation to Negotiate, NTLM.

Setting up SSO by impersonation

September 21, 2020

You can configure the KCD account for NetScaler appliance SSO by impersonation. In this configuration, the NetScaler appliance obtains the user's username and passwordwhen the user authenticates to the authentication server and uses those credentials to impersonate the user to obtain a ticket-granting ticket (TGT). If the user's name is in UPN format, the appliance obtains the user's realm from UPN. Otherwise, it obtains the user's name and realm by extracting it from the SSO domain used during initial authentication, or from the session profile.

When configuring the KCD account, you must set the realm parameter to the realm of the service that the user is accessing. The same realm is also used as the user's realm if the user's realm cannot be obtained from authentication with the NetScaler appliance or from the session profile.

To create the KCD account for SSO by impersonation with a password

At the command prompt, type the following command:

“‘add aaa kcdaccount -realmStr

```
For the variables, substitute the following values:

-  **accountname**. The KCD account name.
-  **realm**. The domain assigned to NetScaler appliance SSO.

### Example

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following command:
```

add aaa kcdAccount kcdaccount1 -keytab kcdvserver.keytab

‘‘‘

For information on configuring Kerberos impersonation through NetScaler GUI, see Citrix Support.

Configuring SSO by Delegation

September 21, 2020

To configure SSO by Delegation, you need to perform the following tasks:

- If you are configuring delegation by delegated user certificate, install the matching CA certificates on the NetScaler appliance and add them to the NetScaler appliance configuration.
- Create the KCD account on the appliance. The appliance uses this account to obtain service tickets for your protected applications.
- Configure the Active Directory server.

Installing the client CA certificate on the NetScaler appliance

If you are configuring NetScaler appliance SSO with a client certificate, you must copy the matching CA certificate for the client certificate domain (the client CA certificate) to the NetScaler appliance, and then install the CA certificate. To copy the client CA certificate, use the file transfer program of your choice to transfer the certificate and private-key file to the NetScaler appliance, and store the files in /nsconfig/ssl.

To install the client CA certificate on the NetScaler appliance

At the command prompt, type the following command:

“‘add ssl certKey -cert [(-key [-password]) | -fipsKey][-inform (DER | PEM)][-expiryMonitor (ENABLED | DISABLED | UNSET) [-notificationPeriod]] [-bundle (YES | NO)]

```
For the variables, substitute the following values:

-   **certkeyName**. A name for the client CA certificate. Must begin with an ASCII alphanumeric or underscore (\_) character, and must consist of from one to thirty-one characters. Allowed characters include the ASCII alphanumerics, underscore, hash (\#), period(.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the certificate-key pair is created. If the name includes one or more spaces, enclose the name in double or single quotation marks (for example, "my cert" or 'my cert').
-   **cert**. Full path name and file name of the X509 certificate file used to form the certificate-key pair. The certificate file must be stored on the NetScaler appliance, in the /nsconfig/ssl/ directory.
-   **key**. Full path name and file name of the file that contains the private key to the X509 certificate file. The key file must be stored on the NetScaler appliance in the /nsconfig/ssl/ directory.
- **password**. If a private key is specified, the passphrase used to encrypt the private key. Use this option to load encrypted private keys in PEM format.
- **fipsKey**. Name of the FIPS key that was created inside the Hardware Security Module (HSM) of a FIPS appliance, or a key that was imported into the HSM.
    > **Note**
    >
    > You can specify either a key or a fipsKey, but not both.
- **inform**. Format of the certificate and private-key files, either PEM or DER.
- **passplain**. Pass phrase used to encrypt the private key. Required when adding an encrypted private-key in PEM format.
- **expiryMonitor**. Configure the NetScaler appliance to issue an alert when the certificate is about to expire. Possible values: ENABLED, DISABLED, UNSET.
- **notificationPeriod**. If expiryMonitor is ENABLED, number of days before the certificate expires to issue an alert.
- **bundle**. Parse the certificate chain as a single file after linking the server certificate to its issuer's certificate within the file. Possible values: YES, NO.

#### Example

The following example adds the specified delegated user certificate customer-cert.pem to the NetScaler appliance configuration along with the key customer-key.pem, and sets the password, certificate format, expiration monitor, and notification period.

To add the delegated user certificate, you would type the following commands:
```

"'add ssl certKey customer -cert "/nsconfig/ssl/customer-cert.pem"
-key "/nsconfig/ssl/customer-key.pem" -password "dontUseDefaultPWs!"
-inform PEM -expiryMonitor ENABLED [-notificationPeriod 14]

```
## Creating the KCD account

If you are configuring NetScaler appliance SSO by delegation, you can configure the KCD account to use the user's log-on name and password, to use the user's log-on name and keytab, or to use the user's client certificate. If you configure SSO with user name and password, the NetScaler appliance uses the delegated user account to obtain a Ticket Granting Ticket (TGT), and then uses the TGT to obtain service tickets for the specific services that each user requests.
```

~~~
If you configure SSO with keytab file, the NetScaler appliance uses the delegated user account and keytab information. If you configure SSO with a delegated user certificate, the NetScaler appliance uses the delegated user certificate.

### To create the KCD account for SSO by delegation with a password

At the command prompt, type the following commands:

<!--NeedCopy-->
```add aaa kcdaccount <accountname> -delegatedUser root -kcdPassword <password> -realmStr <realm>```

For the variables, substitute the following values:

-  **accountname**. A name for the KCD account.
-  **password**. A password for the KCD account.
-  **realm**. The realm of the KCD account, usually the domain for which SSO is active.

#### Example (UPN Format)

To add a KCD account named kcdaccount1 to the NetScaler appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in UPN format (as root), you would type the following commands:

<!--NeedCopy-->
~~~

add aaa kcdaccount kcdaccount1 –delegatedUser root
-kcdPassword password1 -realmStr EXAMPLE.COM

```
#### Example (SPN Format)

To add a KCD account named kcdaccount1 to the NetScaler appliance configuration with a password of password1 and a realm of EXAMPLE.COM, specifying the delegated user account in SPN format, you would type the following commands:

<!--NeedCopy-->
```

add aaa kcdAccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser "host/kcdvserver.example.com" -kcdPassword password1

```
### Creating the KCD account for SSO by delegation with a keytab
```

~~~

If you plan to use a keytab file for authentication, first create the keytab. You can create the keytab file manually by logging onto the AD server and using the ktpass utility, or you can use the NetScaler appliance configuration utility to create a batch script, and then run that script on the AD server to generate the keytab file. Next, use FTP or another file transfer program to transfer the keytab file to the NetScaler appliance and place it in the /nsconfig/krb directory. Finally, configure the KCD account for NetScaler appliance SSO by delegation and provide the path and file name of the keytab file to the NetScaler appliance.

To create the keytab file manually

Log on to the AD server command line and, at the command prompt, type the following command:

<!--NeedCopy-->
```ktpass princ <SPN> ptype KRB5_NT_PRINCIPAL mapuser <DOMAIN><username> pass <password> -out <File_Path>```

For the variables, substitute the following values:

-   **SPN**. The service principal name for the KCD service account.
-   **DOMAIN**. The domain of the Active Directory server.
-   **username**. The KSA account username.
-   **password**. The KSA account password.
-   **path**. The full path name of the directory in which to store the keytab file after it is generated.

To use the NetScaler appliance configuration utility to create a script to generate the keytab file

1. Navigate to **Security > AAA - Application Traffic.**
1. In the data pane, under **Kerberos Constrained Delegation**, click **Batch** file to generate Keytab.
1. In the **Generate KCD (Kerberos Constrained Delegation) Keytab Script** dialog box, set the following parameters:
    -   **Domain User Name**. The KSA account username.
    -   **Domain Password**. The KSA account password.
    -   **Service Principal**. The service principal name for the KSA.
    -   **Output File Name**. The full path and file name to which to save the keytab file on the AD server.
1. Clear the **Create Domain User Account** check box.
1. Click **Generate Script**.
~~~

~~~
1.  Log on to the Active Directory server and open a command line window.
1.  Copy the script from the **Generated Script** window and paste it directly into the Active Directory server command-line window. The keytab is generated and stored in the directory under the file name that you specified as **Output File Name**.
1.  Use the file transfer utility of your choice to copy the keytab file from the Active Directory server to the NetScaler appliance and place it in the /nsconfig/krb directory.

#### To create the KCD account

At the command prompt, type the following command:

<!--NeedCopy-->
```add aaa kcdaccount <accountname>  – keytab <keytab>```

##### Example

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following commands:

<!--NeedCopy-->
~~~

add aaa kcdaccount kcdaccount1 –keytab kcdvserver.keytab

~~~
### To create the KCD account for SSO by delegation with a delegated user cert

At the command prompt, type the following command:

<!--NeedCopy-->
```add aaa kcdaccount <accountname> -realmStr <realm> -delegatedUser <user_nameSPN> -usercert <cert> -cacert <cacert>```

For the variables, substitute the following values:

-  **accountname**. A name for the KCD account.
-  **realmStr**. The realm for the KCD account, usually the domain for which SSO is configured.
-  **delegatedUser**. The delegated user name, in SPN format.
-  **usercert**. The full path and name of the delegated user certificate file on the NetScaler appliance. The delegated user certificate must contain both the client certificate and the private key, and must be in PEM format. If you use smart card
~~~

```
    authentication, you might need to create a smart card certificate template to allow certificates to be imported with the private key.
-  **cacert**. The full path to and name of the CA certificate file on the NetScaler appliance.

#### Example

To add a KCD account named kcdccount1, and use the keytab named kcdvserver.keytab, you would type the following command:

<!--NeedCopy-->
```

```
add aaa kcdaccount kcdaccount1 -realmStr EXAMPLE.COM
-delegatedUser “host/kcdvserver.example.com” -usercert /certs/usercert
-cacert /cacerts/cacert
```

```
## Setting up Active Directory for NetScaler appliance SSO

When you configure SSO by delegation, in addition to creating the KCDAccount on the NetScaler appliance, you must also create a matching Kerberos Service Account (KSA) on your LDAP active directory server, and configure the server for SSO. To create the KSA, use the account creation process on the active directory server. To configure SSO on the active directory server, open the properties window for the KSA. In the Delegation tab, enable the following options: Trust this user for delegation to specified services only and Use any Authentication protocol. (The Kerberos only option does not work, because it does not enable protocol transition or constrained delegation.) Finally, add the services that NetScaler appliance SSO will manage.

> **Note**
>
> If the Delegation tab is not visible in the KSA account properties dialog box, before you can configure the KSA as described, you must use the Microsoft
setspn command-line tool to configure the active directory server so that the tab is visible.

### To configure delegation for the Kerberos service account

1.  In the LDAP account configuration dialog box for the Kerberos service account that you created, click the **Delegation** tab.
1.  Choose "Trust this user for delegation to the specified services only".
```

```
1.  Under "Trust this user for delegation to the specified services only," choose "Use any authentication protocol".
1.  Under “Services to which this account can present delegated credentials,” click **Add.**
1.  In the **Add Services** dialog box, click **Users** or **Computers**, choose the server that hosts the resources to be assigned to the service account, and then click **OK.**
    > **Note**
    >
    > Constrained delegation does not support services hosted in domains other than the domain assigned to the account, even though Kerberos might have a trust relationship with other domains
1.  Back in the **Add Services** dialog box, in the **Available Services** list, choose the services assigned to the service account. NetScaler appliance SSO supports the HTTP and MSSQLSVC services.
1.  Click **OK.**
<!--NeedCopy-->
```

Generating the KCD keytab script

September 21, 2020

The KCD Keytab Script dialog box generates the keytab script, which in turn generates the keytab file necessary to configure KCD on the NetScaler appliance.

To generate the KCD keytab script by using the configuration utility

1. Navigate to **Security > AAA - Application Traffic.**
2. In the details pane, under **Kerberos Constrained Delegation**, click Batch file to generate keytab.
3. In the Generate KCD (Kerberos Constrained Delegation) **Keytab Script** dialog box, fill out the fields as described below.
   - **Domain User Name:** The name of the domain user.
   - **Domain Password:** The password for the domain user.
   - **Service Principal:** The service principal.
   - **Output File Name:** A filename for the KCD script file.
   - **Create Domain User Account:** Select this check box to create the specified domain user account.

4. Click **Generate Script** to generate the script. The script is generated, and appears in the **Generated Script** text box below the **Generate Script** button.
5. Copy the script, and save it as a file on your AD domain controller. You must now run this script on the domain controller to generate the keytab file, and then copy the keytab file to the /nsconfig/krb/ directory on the NetScaler appliance.
6. Click **OK.**

SAML authentication

September 21, 2020

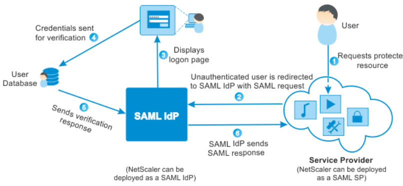
Security Assertion Markup Language (SAML) is an XML-based authentication mechanism that provides single sign-on capability and is defined by the OASIS Security Services Technical Committee.

Why SAML?

Consider a scenario in which a service provider (LargeProvider) hosts a number of applications for a customer (BigCompany). BigCompany has users that must seamlessly access these applications. In a traditional setup, LargeProvider would need to maintain a database of users of BigCompany. This raises some concerns for each of the following stakeholders:

- LargeProvider must ensure security of user data.
- BigCompany must validate the users and keep the user data up-to-date, not just in its own database, but also in the user database maintained by LargeProvider. For example, a user removed from the BigCompany database must also be removed from the LargeProvider database.
- A user has to log on individually to each of the hosted applications.

The SAML authentication mechanism provides an alternative approach. The following deployment diagram shows how SAML works.


Credentials sent
for verification ④
Displays
③ logon page
User
① Requests protected
resource
User
Database
Unauthenticated user is redirected
to SAML IdP with SAML request
②
⑤ Sends verification
response
SAML IdP
⑥
SAML IdP sends
SAML response
Service Provider
(NetScaler can be deployed
as a SAML SP)
(NetScaler can be
deployed as a SAML IdP)

The concerns raised by traditional authentication mechanisms are resolved as follows:

- LargeProvider does not have to maintain a database for BigCompany users. Freed from identity management, LargeProvider can concentrate on providing better services.
- BigCompany does not bear the burden of making sure the LargeProvider user database is kept in sync with its own user database.

- A user can log on once, to one application hosted on LargeProvider, and be automatically logged on to the other applications that are hosted there.

The NetScaler appliance can be deployed as a SAML Service Provider (SP) and a SAML Identity Provider (IdP). Read through the relevant topics to understand the configurations that must be performed on the NetScaler appliance.

A NetScaler appliance configured as a SAML service provider can now enforce an audience restriction check. The audience restriction condition evaluates to "Valid" only if the SAML replying party is a member of at least one of the specified audiences.

You can configure a NetScaler appliance to parse attributes in SAML assertions as group attributes. Parsing them as group attributes enables the appliance to bind policies to the groups.

> **Note**
>
> A NetScaler appliance MPX FIPS appliance used as a SAML service provider now supports encrypted assertions. Also, a NetScaler appliance MPX FIPS appliance functioning as a SAML service provider or a SAML identity provider can now be configured to use the SHA2 algorithms on FIPS hardware.

Configuring FIPS offload support using the command line interface

1. Add SSL FIPS

   add ssl fipsKey fips-key

2. Create a CSR and use it at CA server to generate a certificate. You can then copy the certificate in **/nsconfig/ssl**. Let's assume that the file is *fips3cert.cer*.

   ```
   add ssl certKey fips-cert -cert fips3cert.cer -fipsKey fips-key<!--
   NeedCopy-->
   ```

3. Specify this certificate in the SAML action for SAML SP module

   ```
   set samlAction <name> -samlSigningCertName fips-cert<!--NeedCopy-->
   ```

4. Use the certificate in samlIdpProfile for SAML IDP module

   ```
   set samlidpprofile fipstest -samlIdpCertName fips-cert<!--NeedCopy-->
   ```

The following table lists some articles that are specific to deployments where the NetScaler appliance is used as a SAML SP or a SAML IdP.

SAML SP	SAML IdP	Information Link
NetScaler	Citrix AppController Z3	Citrix Support
NetScaler	CloudGateway	Citrix Support

SAML SP	SAML IdP	Information Link
NetScaler	Microsoft AD FS 2.0	Citrix Support
NetScaler	Shibboleth	Citrix Support
NetScaler	Shibboleth (With SAML single logout configuration)	Citrix Support
Siteminder	NetScaler	Citrix Support
ShareFile	NetScaler	Citrix Support

NetScaler appliance as a SAML SP

June 9, 2021

The SAML Service Provider (SP) is a SAML entity that is deployed by the service provider. When a user tries to access a protected application, the SP evaluates the client request. If the client is unauthenticated (does not have a valid NSC_TMAA or NSC_TMAS cookie), the SP redirects the request to the SAML Identity Provider (IdP).

The SP also validates SAML assertions that are received from the IdP.

When the NetScaler appliance is configured as an SP, all user requests are received by a traffic management virtual server (load balancing or content switching) that is associated with the relevant SAML action.

The NetScaler appliance also supports POST and Redirect bindings during logout.

> **Note**
>
> A NetScaler appliance can be used as a SAML SP in a deployment where the SAML IdP is configured either on the appliance or on any external SAML IdP.

When used as a SAML SP, a NetScaler appliance:

- Can extract the user information (attributes) from the SAML token. This information can then be used in the policies that are configured on the NetScaler appliance. For example, if you want to extract the GroupMember and emailaddress attributes, in the SAMLAction, specify the **Attribute2** parameter as GroupMember and the **Attribute3** parameter as emailaddress.

  > **Note**
  >
  > Default attributes such as username, password, and logout URL must not be extracted in attributes 1 to 16, because they as are implicitly parsed and stored in the session.

- Can extract attribute names of upto 127 bytes from an incoming SAML assertion. The previous limit was 63 bytes. Support introduced in NetScaler 11.0 Build 64.x.
- Supports post, redirect, and artifact bindings. Support for redirect and artifact bindings is introduced in NetScaler 11.0 Build 55.x.

  > **Note**
  >
  > Redirect binding should not be used for large amount of data, when the assertion after inflate or decoding is greater than 10K.

- Can decrypt assertions. Support introduced in NetScaler 11.0 Build 55.x.
- Can extract multi-valued attributes from a SAML assertion. These attributes are sent is nested XML tags such as:

  <AttributeValue> <AttributeValue>Value1</AttributeValue>
  <AttributeValue>Value2</AttributeValue>
  </AttributeValue>

  When presented with above XML, the NetScaler appliance can extract both Value1 and Value2 as values of a given attribute, as opposed to the old firmware that extracts only Value1.

  > **Note**
  >
  > Support introduced in NetScaler 11.0 Build 64.x.

- Can specify the validity of a SAML assertion.

  If the system time on NetScaler appliance SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions will be valid.

  This duration, called the "skew time," specifies the number of minutes for which the message should be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

  > **Note**
  >
  > Support introduced in NetScaler 11.0 Build 64.x.

- Can send additional attribute called 'ForceAuth' in the authentication request to external IDP (Identity Provider). By default, the ForceAuthn is set to 'False'. It can be set to 'True' to suggest IDP to force authentication despite existing authentication context. Additionally, NetScaler appliance SP does authentication request in query parameter when configured with artifact binding.

To configure the NetScaler appliance as a SAML SP by using the command line interface

1. Configure a SAML SP action.

Example

The following command adds a SAML action that redirects unauthenticated user requests.

```
add authentication samlAction SamlSPAct1 -samlIdPCertName nssp -samlRedirectUrl
 https://auth1.example.com -relaystateRule "AAA.LOGIN.RELAYSTATE.EQ(\"
https://lb.example1.com/\")"
```

Points to note

- Certificate provided for `-samlIdPCertName` in the samlAction command must match the corresponding certificate from IdP for the signature verification to succeed.
- Citrix recommends to have a full domain name with trailing '/' in the expression.
- Administrators must configure an expression for **relaysStateRule** in the samlAction command. The expression must contain the list of published domains that the user connects to before being redirected to the authentication virtual server. For example, the expression must contain the domains of the front-end virtual server (VPN, LB, or CS) that use this SAML action for authentication.

For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction and https://support.citrix.com/article/CTX316577.

2. Configure the SAML policy.

   Example

   The following command defines a SAML policy that applies the above defined SAML action to all traffic.

   ```
   add authentication samlPolicy SamlSPPol1 ns_true SamlSPAct1
   ```

3. Bind the SAML policy to the authentication virtual server.

   Example

   The following command binds the SAML policy to a authentication virtual server named "av_saml".

   ```
   bind authentication vserver av_saml -policy SamlSPPol1
   ```

4. Bind the authentication virtual server to the appropriate traffic management virtual server.

   Example

   The following command adds a load balancing virtual server named "lb1_ssl" and associates the authentication virtual server named "av_saml" to the load balancing virtual server.

   ```
   add lb vserver lb1_ssl SSL 10.217.28.224 443 -persistenceType NONE -
   cltTimeout 180 -AuthenticationHost auth1.example.com -Authentication ON
    -authnVsName av_saml
   ```

To configure a NetScaler appliance as a SAML SP by using the graphical user interface

1. Configure the SAML action and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, create a policy with SAML as the action type, and associate the required SAML action with the policy.

2. Associate the SAML policy with an authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the SAML policy with the authentication virtual server.

3. Associate the authentication server with the appropriate traffic management virtual server.

   Navigate to **Traffic Management > Load Balancing** (or **Content Switching**) > **Virtual Servers**, select the virtual server, and associate the authentication virtual server with it.

NetScaler appliance as a SAML IdP

June 9, 2021

The SAML IdP (Identity Provider) is a SAML entity that is deployed on the customer network. The IdP receives requests from the SAML SP and redirects users to a logon page, where they must enter their credentials. The IdP authenticates these credentials with the user directory (external authentication server, such as LDAP) and then generates a SAML assertion that is sent to the SP.

The SP validates the token, and the user is then granted access to the requested protected application.

When the NetScaler appliance is configured as an IdP, all requests are received by an authentication virtual server that is associated with the relevant SAML IdP profile.

> **Note**
>
> A NetScaler appliance can be used as a IdP in a deployment where the SAML SP is configured either on the appliance or on any external SAML SP.

When used as a SAML IdP, a NetScaler appliance:

- Supports all authentication methods that it supports for traditional logons.
- Digitally signs assertions. Support for the SHA256 algorithm is introduced in NetScaler 11.0 Build 55.x.
- Supports single-factor and two-factor authentication. SAML must not be configured as the secondary authentication mechanism.

- Can encrypt assertions by using the public key of the SAML SP. This is recommended when the assertion includes sensitive information. Support introduced in NetScaler 11.0 Build 55.x.
- Can be configured to accept only digitally signed requests from the SAML SP. Support introduced in NetScaler 11.0 Build 55.x.
- Can log on to the SAML IdP by using the following 401-based authentication mechanisms: Negotiate, NTLM, and Certificate. Support introduced in NetScaler 11.0 Build 55.x.
- Can be configured to send 16 attributes in addition to the NameId attribute. The attributes must be extracted from the appropriate authentication server. For each of them, you can specify the name, the expression, the format, and a friendly name in the SAML IdP profile. Support introduced in NetScaler 11.0 Build 55.x.
- If the NetScaler appliance is configured as a SAML IdP for multiple SAML SP, a user can gain access to applications on the different SPs without explicitly authenticating every time. The NetScaler appliance creates a session cookie for the first authentication, and every subsequent request uses this cookie for authentication. Support introduced in NetScaler 11.0 Build 55.x.
- Can send multi-valued attributes in a SAML assertion. Support introduced in NetScaler 11.0 Build 64.x.
- Supports post and redirect bindings. Support for redirect bindings is introduced in NetScaler 11.0 Build 64.x.
- Can specify the validity of a SAML assertion.

  If the system time on NetScaler appliance SAML IdP and the peer SAML SP is not in sync, the messages might get invalidated by either party. To avoid such cases, you can now configure the time duration for which the assertions will be valid.

  This duration, called the "skew time," specifies the number of minutes for which the message should be accepted. The skew time can be configured on the SAML SP and the SAML IdP.

  > **Note**
  >
  > Support introduced in NetScaler 11.0 Build 64.x.

- Can be configured to serve assertions only to SAML SPs that are pre-configured on or trusted by the IdP. For this configuration, the SAML IdP must have the service provider ID (or issuer name) of the relevant SAML SPs. Support introduced in NetScaler 11.0 Build 64.x.

  > **Note**
  >
  > Before proceeding, make sure that you have an authentication virtual server that is linked to an LDAP authentication server.

To configure a NetScaler appliance as a SAML IdP by using the command line interface

1. Configure a SAML IdP profile.

   Example

   Adding Citrix ADC appliance as an IdP with SiteMinder as the SP.

   ```
   add authentication samlIdPProfile samlIDPProf1 -samlSPCertName siteminder
   -cert -encryptAssertion ON -samlIdPCertName ns-cert -assertionConsumerServiceURL
    http://sm-proxy.nsi-test.com:8080/affwebservices/public/saml2assertionconsumer
    -rejectUnsignedRequests ON -signatureAlg RSA-SHA256 -digestMethod
   SHA256 -acsUrlRule AAA.LOGIN.SAML_REQ_ACS_URL.REGEX_MATCH(re##^https://
   example2\.com/cgi/samlauth$##)
   ```

   Points to note

   - In SAML IdP profile, configure **acsURLRule** that takes an expression of the list of applicable service provider URLs for this IdP. This expression depends on the SP being used. If Citrix ADC is configured as SP, ACS URL will be `https://<SP-domain_name>/cgi/samlauth`. Citrix recommends having a full URL in the expression for matching.
   - You must specify the starting of the domain with "^" sign (example: ^https) along with the dollar sign "$" at the end of the string (example: samlauth$).

   For more details on the command, see https://developer-docs.citrix.com/projects/citrix-adc-command-reference/en/latest/authentication/authentication-samlAction and https://support.citrix.com/article/CTX316577.

2. Configure the SAML authentication policy and associate the SAML IdP profile as the action of the policy.

   ```
   add authentication samlIdPPolicy samlIDPPol1 -rule ns\\_true -action
   samlIDPProf1
   ```

3. Bind the policy to the authentication virtual server.

   ```
   bind authentication vserver saml-auth-vserver -policy samlIDPPol1 -
   priority 100
   ```

To configure a NetScaler appliance as a SAML IdP by using the graphical user interface

1. Configure the SAML IdP profile and policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with SAML IdP as the action type, and associate the required SAML IdP profile with the policy.

2. Associate the SAML IdP policy with an authentication virtual server.

   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the SAML IdP policy with the authentication virtual server.

Configuring SAML single sign-on

September 21, 2020

To provide single sign-on capabilities across applications that are hosted on the service provider, you can configure SAML single sign-on on the SAML SP.

Configuring SAML single sign-on by using the command line interface

1. Configure the SAML SSO profile.

   Example

   In the following command, Example is the load balancing virtual server that has a web link from the SharePoint portal. Nssp.example.com is the Traffic Management virtual server that is load balancing the SharePoint server.

   ```
   add tm samlSSOProfile tm-saml-sso -samlSigningCertName nssp -assertionConsumerSer
    "https://nssp2.example.com/cgi/samlauth"-relaystateRule "\\"https://
   nssp2.example.com/samlsso.html\\"" -sendPassword ON -samlIssuerName
   nssp.example.com<!--NeedCopy-->
   ```

2. Associate the SAML SSO profile with the traffic action.

   Example

   The following command enables SSO and binds the SAML SSO profile created above to a traffic action.

   ```
   add tm trafficAction html\\_act -SSO ON -samlSSOProfile tm-saml-sso<!--
   NeedCopy-->
   ```

3. Configure the traffic policy that specifies when the action must be executed.

   Example

   The following command associates the traffic action with a traffic policy.

   ```
   add tm trafficPolicy html_pol "HTTP.REQ.URL.CONTAINS(\\"abc.html\\")"
   html_act<!--NeedCopy-->
   ```

4. Bind the traffic policy created above to a traffic management virtual server (load balancing or content switching). Alternatively, the traffic policy can be associated globally.

> **Note**
>
> This traffic management virtual server must be associated with the relevant authentication virtual sever that is associated with the SAML action.

```
bind lb vserver lb1_ssl -policyName html_pol -priority 100 -gotoPriorityExpressio
 END -type REQUEST<!--NeedCopy-->
```

Configuring SAML single sign-on by using the graphical user interface

1. Define the SAML SSO profile, the traffic profile, and the traffic policy.

   Navigate to **Security > AAA - Application Traffic > Policies > Traffic**, select the appropriate tab, and configure the settings.

2. Bind the traffic policy to a traffic management virtual server or globally to the NetScaler appliance.

OAuth authentication

September 21, 2020

The AAA traffic management feature now supports OAuth and OpenID-Connect mechanisms for authenticating and authorizing users to applications that are hosted on applications such as Google, Facebook, and Twitter.

The authentication mehanism facilitates the inline verification of OpenID tokens. The NetScaler appliance can be configured to obtain certificates and verify signatures on the token.

A major advantage of using the OAuth and OpenID-Connect mechanisms is that the user information is not sent to the hosted applications and therefore the risk of identity theft is considerably reduced.

The NetScaler appliance configured for AAA now accepts incoming tokens that are signed using HMAC HS256 algorithm. In addition, the public keys of the SAML Identity Provider (IdP) are read from a file, instead of learning from an URL endpoint.

In the NetScaler appliance implementation, the application to be accessed is represented by the AAA traffic management virtual server. So, to configure OAuth, you must configure an OAuth policy which must then be associated with a AAA traffic management virtual server.

> **Note**
>
> OAuth on NetScaler appliance is currently qualified for all SAML IdPs that are compliant with "OpenID connect 2.0".

To configure OAuth by using the configuration utilty:

1. Configure the OAuth action and policy.
   Navigate to **Security > AAA - Application Traffic > Policies > Authentication > Advanced Policies > Policy**, and create a policy with OAuth as the action type, and associate the required OAuth action with the policy.
2. Associate the OAuth policy with an authentication virtual server.
   Navigate to **Security > AAA - Application Traffic > Virtual Servers**, and associate the OAuth policy with the authentication virtual server.

> **Note**
>
> Attributes (1 to 16) can be extracted in the OAuth response. Currently these attributes are not evalauted. They are added for the future reference.

To configure OAuth by using the command line interface:

1. Define an OAuth action.

   ```
   add authentication OAuthAction <name> -authorizationEndpoint <URL>
    -tokenEndpoint <URL> [-idtokenDecryptEndpoint <URL>] -clientID <
   string> -clientSecret <string> [-defaultAuthenticationGroup <string
   >][-tenantID <string>][-GraphEndpoint <string>][-refreshInterval
   <positive_integer>] [-CertEndpoint <string>][-audience <string>][-
   userNameField <string>][-skewTime <mins>][-issuer <string>][-Attribute1
    <string>][-Attribute2 <string>][-Attribute3 <string>]...<!--NeedCopy
   -->
   ```
2. Associate the action with an advanced authentication policy.
   "'add authentication Policy** -rule -action

   ~~~~
   **Example**
   ```preblock
   add authentication oauthAction a -authorizationEndpoint https://
       example.com/ -tokenEndpoint https://example.com/ -clientiD sadf
        -clientsecret df<!--NeedCopy-->
   ~~~~

For more information on authentication OAuthAction parameters, see "authentication OAuthAction".

> **Note**
>
> When a certEndpoint is specified, the NetScaler appliance polls that endpoint at the configured frequency to learn the keys. To configure a NetScaler appliance to read the local file and parse keys from that file, a new configuration option is introduced as follows.

```
set authentication OAuthAction <> -**CertFilePath** <path to local file
with jwks><!--NeedCopy-->
```

Multi-Factor (nFactor) authentication

September 21, 2020

> **Important**
>
> Supported from NetScaler 11.0 Build 62.x onwards.

Multi-factor authentication enhances the security of an application by requiring users to provide multiple proofs of identify to gain access. The NetScaler appliance provides an extensible and flexible approach to configuring multi-factor authentication. This approach is called *nFactor authentication*.

With nFactor authentication you can:

- Configure any number of authentication factors.
- Base the selection of the next factor on the result of executing the previous factor.
- Customize the login interface. For example, you can customize the label names, error messages, and help text.
- Extract user group information without doing authentication.
- Configure pass-through for an authentication factor. This means that no explicit login interaction is required for that factor.
- Configure the order in which different types of authentication are applied. Any of the authentication mechanisms that are supported on the NetScaler appliance can be configured as any factor of the nFactor authentication setup. These factors are executed in the order in which they are configured.
- Configure the NetScaler appliance to proceed to an authentication factor that must be executed when authentication fails. To do so, you configure another authentication policy with the exact same condition, but with the next highest priority and with the action set to "NO_AUTH". You must also configure the next factor, which must specify the alternative authentication mechanism to apply.

For a nFactor benefit, see One Public IP for AAA-TM Deployments on NetScaler.

Sample deployments using nFactor authentication:

- Getting two passwords up-front, pass-through in next factor. Read
- Group extraction followed by certificate or LDAP authentication, based on group membership. Read
- SAML followed by LDAP or certificate authentication, based on attributes extracted during SAML. Read
- SAML in first factor, followed by group extraction, and then LDAP or certificate authentication, based on groups extracted. Read
- Prefilling user name from certificate. Read

- Certificate authentication followed by group extraction for 401 enabled traffic management virtual servers. Read
- Username and 2 passwords with group extraction in third factor. Read
- Certificate fallback to LDAP in same cascade; one virtual server for both certificate and LDAP authentication. Read
- LDAP in first factor and WebAuth in second factor. Read
- Domain drop down in first factor, then different policy evaluations based on group. Read

How to configure nFactor authentication

September 21, 2020

The primary entity used for nFactor authentication is called a *login schema*.

A login schema specifies an authentication schema XML file that defines the manner in which the login form will be rendered. Considering the interaction that the user must have when logging in to the application, you can create a single file for multiple factors or different files for different factors. View sample XML file.

- **Single file for multiple factors.** User will be provided a single form in which to provide credentials for multiple authentication factors.

- **Different files for different factors.** User will be provided a different form for each authentication factor.

Next, you must associate the XML file(s) with login schema(s). You can also specify expressions to extract the user name and the password from the login form.

> **Tip**
>
> You can configure an authentication factor to be pass-through. This means that the user is not required to provide credentials explicitly and there is no login form for that factor. The credentials are either taken from the previous factor or the user name and/or password are dynamically extracted by using the username/password expressions that are configured for that login schema. You must set the login schema to "NOSCHEMA", instead of an XML file.

Now that the login schemas are configured, you must specify the manner in which they must be invoked. A login schema can be invoked by using either a login schema policy or an authentication policy label. The decision depends on the following:

- **Login schema policy.**
  - Specifies the condition on which the login form must be presented to the user.
  - Must be bound to an authentication virtual server.

  - In an authentication virtual server that has multiple login schema policies, the policy with the highest priority that evaluates to true is executed. That is, the login form associated with that policy is presented to the user.
  - The login schema policy is only used to present the first login form.
- **Authentication policy label.**
  - Specifies a collection of authentication policies for a particular factor. Each policy label corresponds to a single factor.
  - Specifies the login form that must be presented to the user.
  - Must be bound as the next factor of an authentication policy or of another authentication policy label.
  - Typically, a policy label includes authentication policies for a specific authentication mechanism. However, you can also have a policy label that has authentication policies for different authentication mechanisms.

To summarize, the configurations you must perform to set up nFactor authentication are as follows:

1. Create the authentication schema XML files.
2. Associate each XML file with a login schema.
3. Associate each login schema with a login schema policy or authentication policy label.
4. Bind login schema policy to an authentication virtual server.
5. Bind authentication policy label, as next factor, to an authentication policy.

How nFactor authentication works

September 21, 2020

Imagine a user requesting access to an application that requires user credentials. As is the case in NetScaler appliance deployments, the request arrives at the NetScaler appliance through a traffic management virtual server (in this case, a load balancing virtual server). Since the user must provide authentication credentials, the load balancing virtual server redirects the request to the authentication virtual server, which does the following:

1. Checks to determine whether any login schema policies are associated with the authentication virtual server.

   - If yes, the user is presented the login form associated with the login schema policy with the highest priority that evaluates to true.

   - If no, the default login form is presented to the user.

   > **Note**
   >
   > The default login schema files are available in the */nsconfig/loginschema/LoginSchema/*

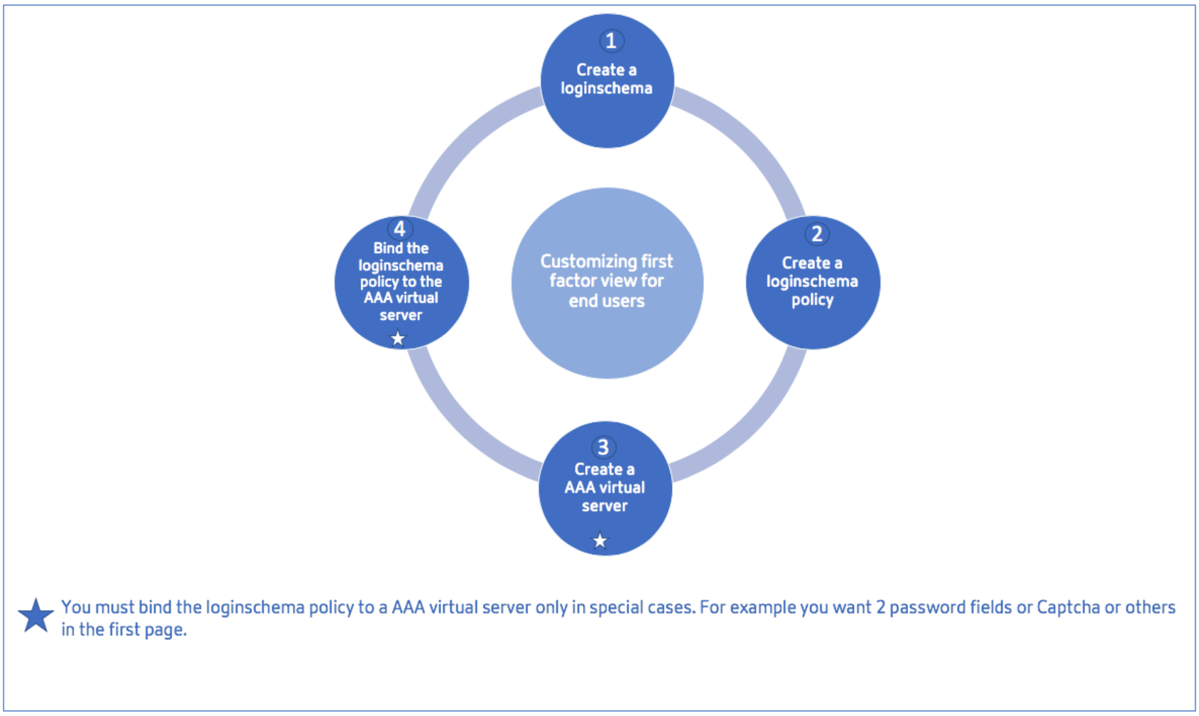
> directory of the NetScaler appliance. Citrix recommends that you copy these files to the */nsconfig/loginschema/* directory before using them, so that changes made to the files are preserved post reboot.

2. The authentication policies that are associated with the authentication virtual server are evaluated. For the policies that are evaluated to true, the actions are executed in order of priority until one of the actions succeeds.
3. The policy label that is associated as the next factor is invoked.
4. The authentication policies that are associated with the authentication policy label are evaluated. For the policies that are evaluated to true, the actions are executed in order of priority until one of the actions succeeds.
5. The policy label that is associated as the next factor is invoked.
6. Steps 4 and 5 are performed repetitively till all the configured next factors are executed.

Configuring nFactor authentication

September 21, 2020

The following diagram illustrates the high-level steps involved in nfactor configuration.


1 Create a loginschema
2 Create a loginschema policy
3 Create a AAA virtual server ★
4 Bind the loginschema policy to the AAA virtual server ★
Customizing first factor view for end users
★ You must bind the loginschema policy to a AAA virtual server only in special cases. For example you want 2 password fields or Captcha or others in the first page.

To understand the step-wise configurations for nFactor authentication, let us consider a 2-factor authentication deployment where the 1st factor is LDAP authentication and the 2nd factor is RADIUS authentication.

This sample deployment requires the user to login to both factors using a single login form. Therefore, we define a single login form that accepts two passwords. The first password is used for LDAP authentication and the other for RADIUS authentication. View XML file.

Here are the configurations that are performed.

1. Configure the load balancing virtual server for authentication

   ```
   add lb vserver lbvs55 HTTP 1.217.193.55 80 -AuthenticationHost auth56.aaatm.com -Authentication ON
   ```

2. Configure the authentication virtual server.

   ```
   add authentication vserver auth56 SSL 1.217.193.56 443 -AuthenticationDomain aaatm.com
   ```

3. Configure the login schema for the login form and bind it to a login schema policy.

   ```
   add authentication loginSchema login1 -authenticationSchema login-2passwd.xml -userCredentialIndex 1 -passwordCredentialIndex 2
   ```

   ```
   add authentication loginSchemaPolicy login1 -rule true -action login1
   ```

4. Configure a login schema for the pass-through and bind it to a policy label

   ```
   add authentication loginSchema login2 -authenticationSchema noschema
   ```

   ```
   add authentication policylabel label1 -loginSchema login2
   ```

5. Configure the LDAP and RADIUS policies.

   ```
   add authentication ldapAction ldapAct1 -serverIP 1.217.28.180 -ldapBase “dc=aaatm, dc=com” -ldapBindDn administrator@aaatm.com -ldapBindDnPassword 71ca2b11ad800ce2787fb7deb5484
   -encrypted -encryptmethod ENCMTHD_3 -ldapLoginName samAccountName -groupAttrName memberOf -subAttributeName CN
   ```

   ```
   add authentication Policy ldap -rule true -action ldapAct1
   ```

   ```
   add authentication radiusAction radius -serverIP 1.217.22.20 -radKey a740d6a0aeb3288fa0a6fbe932d329a
   -encrypted -encryptmethod ENCMTHD_3 -radNASip ENABLED -radNASid NS28.50 -radAttributeType 11 -ipAttributeType 8
   ```

   ```
   add authentication Policy radius -rule true -action radius
   ```

6. Bind the login schema policy to the authentication virtual server

   ```
   bind authentication vserver auth56 -policy login1 -priority 1 -gotoPriorityExpression END
   ```

7. Bind the LDAP policy (first factor) to the authentication virtual server.

   ```
   bind authentication vserver auth56 -policy ldap -priority 1 -nextFactor label1 -gotoPriorityExpression next
   ```

8. Bind the RADIUS policy (second factor) to the authentication policy label.

   bind authentication policylabel label1 -policyName radius -priority 2 -gotoPriorityExpression end

Configuration support for SameSite cookie attribute

September 21, 2020

The SameSite attribute indicates the browser whether the cookie can be used for cross-site context or only for same-site context. Also, if an application intends to be accessed in cross-site context then it can do so only via HTTPS connection. For details, see RFC6265.

Until Feb 2020, the SameSite attribute was not explicitly set in Citrix ADC. The browser took the default value (None). The non-setting of SameSite attribute did not impact the Citrix Gateway and Citrix ADC AAA deployments.

With certain browsers upgrade, such as Google Chrome 80, there is a change in the default cross-domain behavior of cookies. The SameSite attribute can be set to one of the following values. Default value for Google Chrome is set to Lax. For certain version of other browsers, the default value for SameSite attribute might still be set to None.

- **None**: Indicates the browser to use cookie in cross-site context only on secure connections.
- **Lax**: Indicates the browser to use cookie for requests on the same-site context. In cross-site context, only safe HTTP methods like GET request can use the cookie.
- **Strict**: Use the cookie only in the same site context.

If there is no SameSite attribute in the cookie, the Google Chrome assumes the functionality of SameSite = Lax.
As a result, for deployments within an iframe with cross-site context that require cookies to be inserted by the browser, Google Chrome does not share cross site cookies. As a result, the iframe within the website might not load.

Configure SameSite cookie attribute

A new cookie attribute named SameSite is added to the VPN and Citrix ADC AAA virtual servers. This attribute can be set at the global level and at the virtual server level.

To configure SameSite attribute, you must perform the following:

1. Set the SameSite attribute for the virtual server
2. Bind cookies to the patset (if the browser drops cross-site cookies)

Setting the SameSite attribute by using the CLI

To set the SameSite attribute at the virtual server level, use the following commands.

```
set vpn vserver VP1 -SameSite  [ STRICT | LAX | None ]
set aaa vserver VP1 -SameSite  [ STRICT | LAX | None ]
<!--NeedCopy-->
```

To set the SameSite attribute at the global level, use the following commands.

```
set vpn param VP1 -SameSite  [ STRICT | LAX | None ]
set aaa param VP1 -SameSite  [ STRICT | LAX | None ]
<!--NeedCopy-->
```

Note: The virtual server level setting takes preference over the global level setting. Citrix recommends setting the SameSite cookie attribute at the virtual server level.

Binding cookies to the patset by using the CLI

If the browser drops cross-site cookies, you can bind that cookie string to the existing ns_cookies_SameSite patset so that the SameSite attribute is added to the cookie.

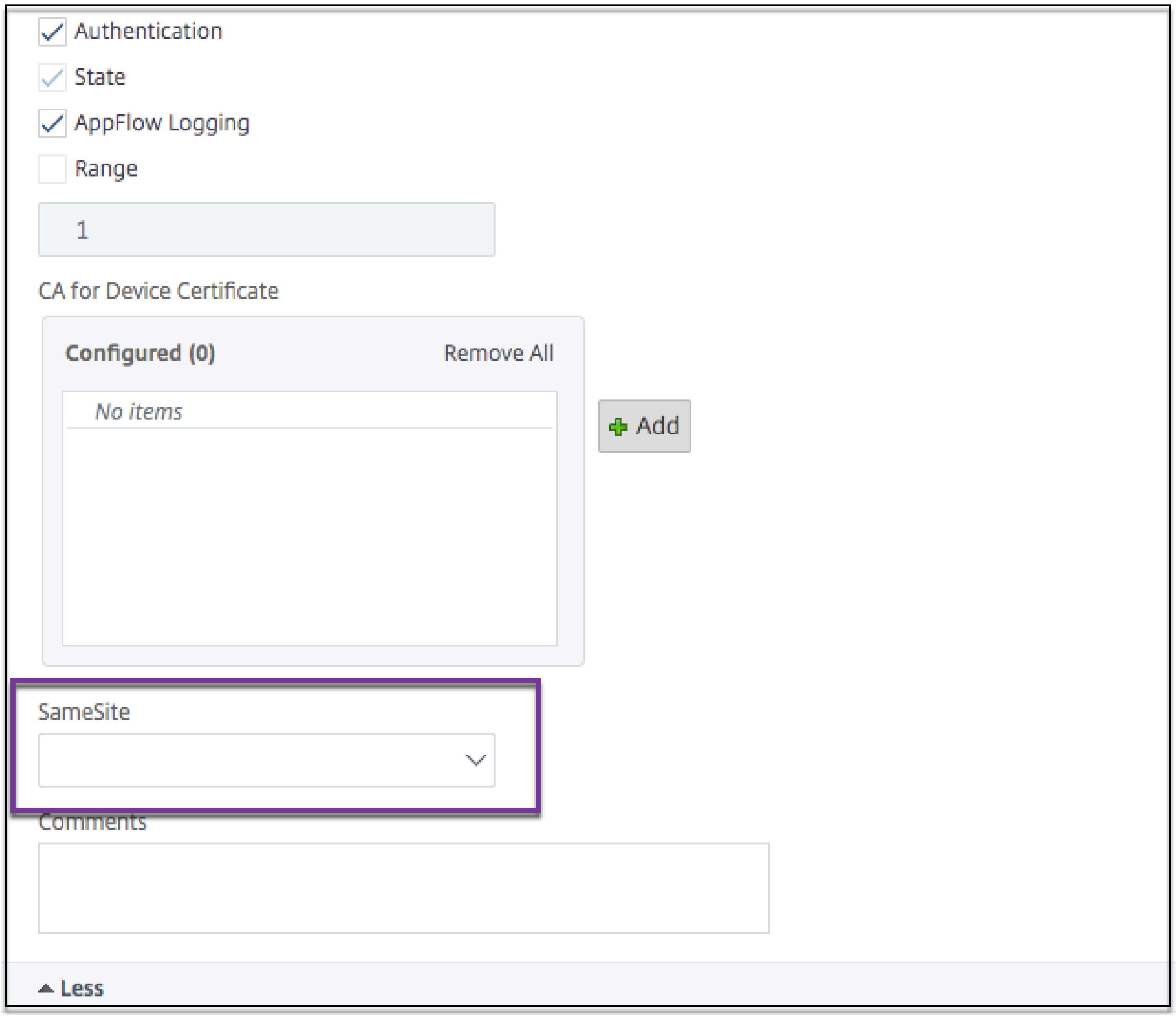
Example:

```
bind patset ns_cookies_SameSite "NSC_TASS"
bind patset ns_cookies_SameSite "NSC_TMAS"
<!--NeedCopy-->
```

Setting the SameSite attribute by using the GUI

To set the SameSite attribute at the virtual server level:

1. Navigate to **Security > AAA – Application Traffic > Virtual Servers.**
2. Select a virtual server and click **Edit**.
3. Click the edit icon in the **Basic Setting**s section and click **More**.
4. In **SameSite**, select the option as required.



☑ Authentication

☑ State

☑ AppFlow Logging

☐ Range

1

CA for Device Certificate

Configured (0) Remove All

No items

Add

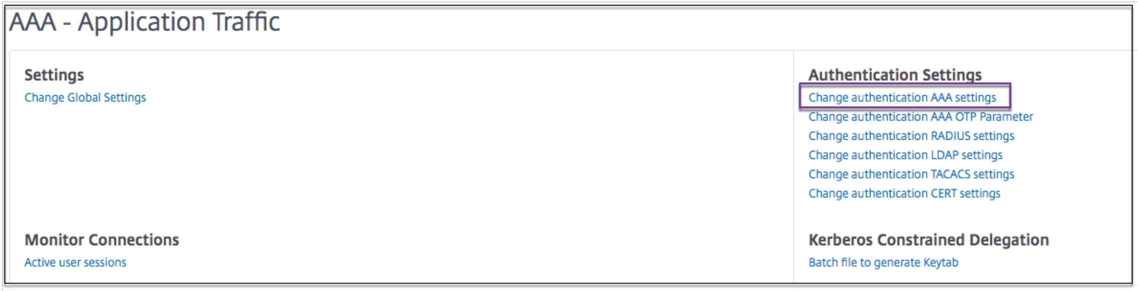
SameSite

Comments

Less

To set the SameSite attribute at the global level:

1. Navigate to **Security > AAA – Application Traffic > Change Authentication Settings.**



AAA - Application Traffic

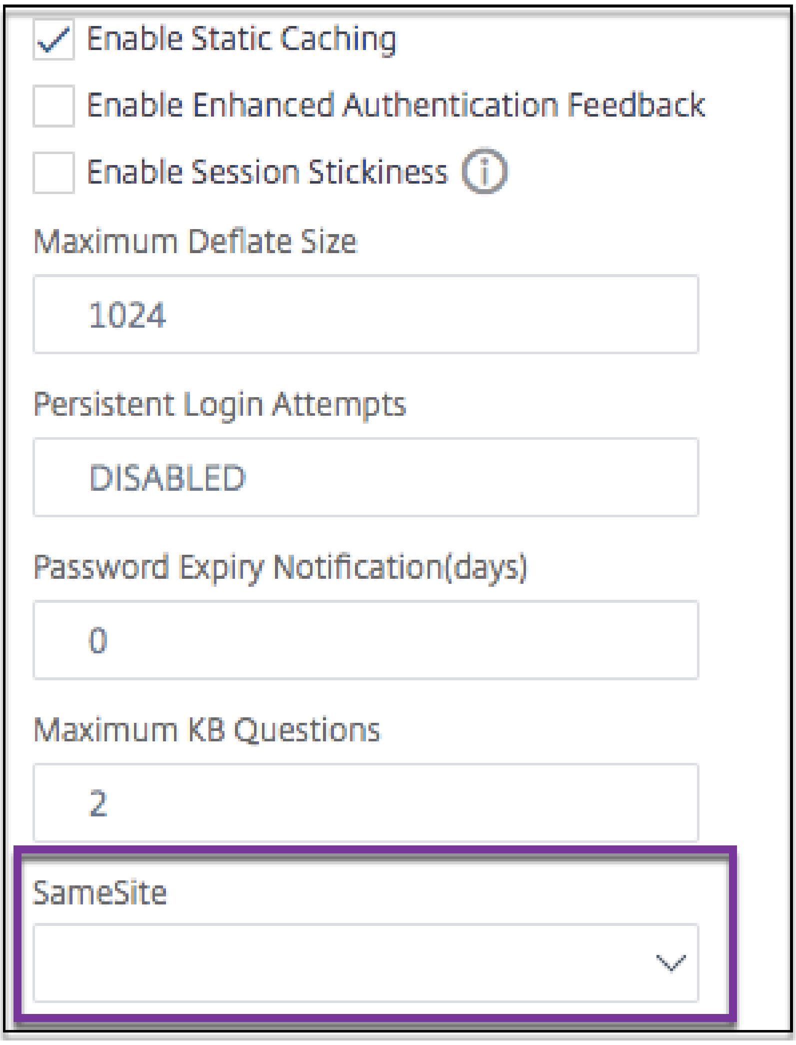
Settings
Change Global Settings

Authentication Settings
Change authentication AAA settings
Change authentication AAA OTP Parameter
Change authentication RADIUS settings
Change authentication LDAP settings
Change authentication TACACS settings
Change authentication CERT settings

Monitor Connections
Active user sessions

Kerberos Constrained Delegation
Batch file to generate Keytab

2. In the **Configure AAA Parameter** page, click the **SameSite** list, and select the option as required.



☑ Enable Static Caching

☐ Enable Enhanced Authentication Feedback

☐ Enable Session Stickiness ⓘ

Maximum Deflate Size

1024

Persistent Login Attempts

DISABLED

Password Expiry Notification(days)

0

Maximum KB Questions

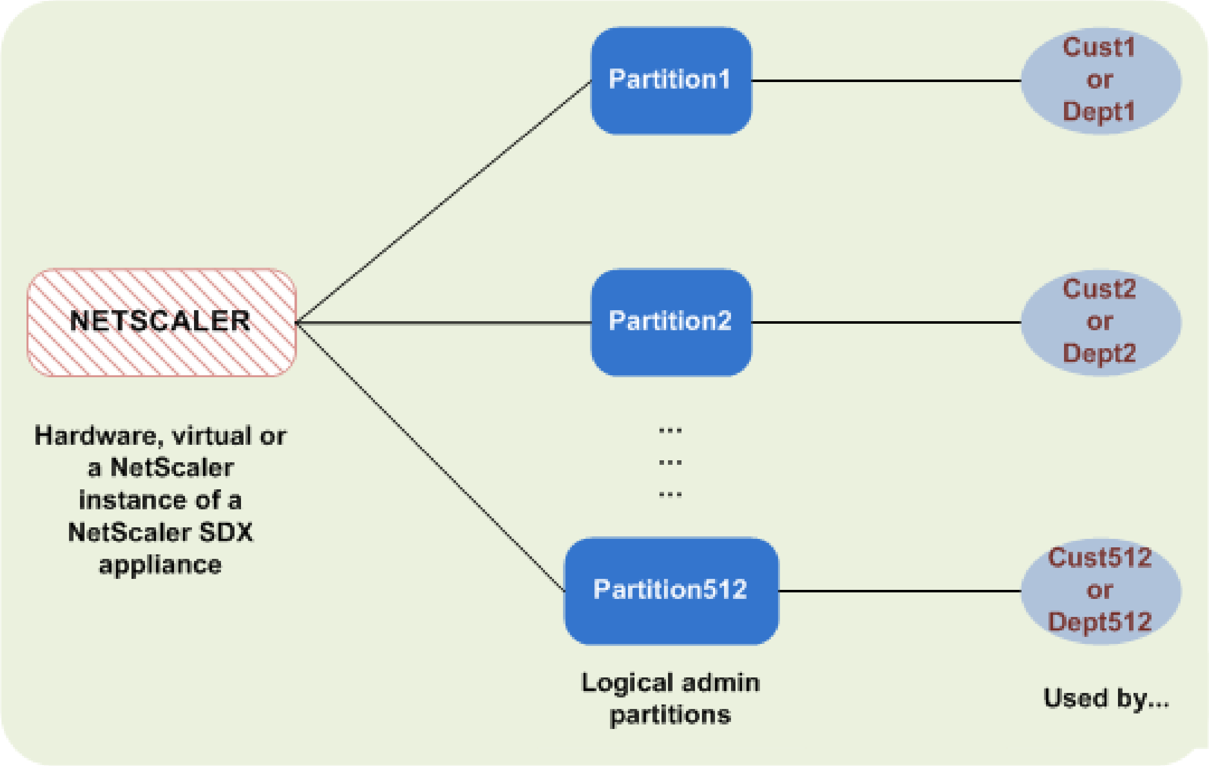
2

SameSite

Admin Partitioning

September 21, 2020

A NetScaler appliance can be partitioned into logical entities called admin partitions, where each partition can be configured and used as a separate NetScaler appliance. The following figure shows the partitions of a NetScaler being used by different customers and departments:


NETSCALER
Hardware, virtual or a NetScaler instance of a NetScaler SDX appliance
Partition1
Partition2
...
...
...
Partition512
Logical admin partitions
Cust1 or Dept1
Cust2 or Dept2
Cust512 or Dept512
Used by...

A partitioned NetScaler appliance has a single default partition and one or more admin partitions. The following table provides further details on the two partition types:

> **Note**
>
> In a partitioned appliance, the mode BridgeBPDUs can be enabled only in the default partition and not in the administrative partitions.

Default Partition

Admin Partitions

Availability:

The NetScaler appliance ships with a single partition, which is called a default partition. The default partition is retained even after the NetScaler appliance is partitioned.

Must be explicitly created as described in Configure admin partitions.

Number of Partitions:

One

A NetScaler appliance can have one or more (maximum of 512) admin partitions.

User Access and Roles:

The default partition can be accessed and configured by all NetScaler users who are not associated with a *partition-specific* command policy. As always, the operations that a user can perform are restricted by the associated command policy.

Can be created only by NetScaler superusers who also specify the users for that partition. Only superusers and associated users of the partition can access and configure the admin partition.

Note: Partition users do not have shell access.

File Structure:

All files in a default partition are stored in the default NetScaler file structure.

For example, the NetScaler configuration file is stored in the */nsconfig* directory and NetScaler logs are stored in the */var/log/* directory.

All files in an admin partition are stored in directory paths that have the name of the admin partition.

For example, the NetScaler configuration file (ns.conf) is stored in the */nsconfig/partitions/<partitionName>* directory. Other partition-specific files are stored in the */var/partitions/<partitionName>* directories.

Some other paths in an admin partition:

- Downloaded files: */var/partitions/<partitionName>/download/*
- Log files: */var/partitions/<partitionName>/log/*

Note: Currently, logging is not supported at partition-level. Therefore, this directory is empty and all logs are stored in the */var/log/* directory.

- SSL CRL certificate related files: */var/partitions/<partitionName>/netscaler/ssl*

Resources Available:

All NetScaler resources.

NetScaler resources that are explicitly assigned to the admin partition.

User access and roles

In authenticating and authorizing a partitioned NetScaler appliance, a root administrator can assign a partition administrator to one or more partitions. The partition administrator can authorize users to that partition without affecting other partitions. These are partition users and they are authorized to access only that partition using SNIP address. Both the root administrator and the partition administrator can configure role based access (RBA by authorizing users to access different applications.

Administrators and user roles can be described as follows:

Root Administrator. Accesses the partitioned appliance through its NSIP address and can grant user access to one or more partitions. The administrator can also assign partition administrators to one or more partitions. The administrator can create a partition administrator from the default partition using a NSIP address or switch to a partition and then create a user and assign partition admin access using a SNIP address.

Partition Administrator. Accesses the specified partition through a NSIP address assigned by the root administrator. The administrator can assign role-based access to partition user access to that partition and also configure external server authentication using partition specific configuration.

System User. Accesses partitions through the NSIP address. Has access to the partitions and resources specified by the root administrator.

Partition User. Accesses a partition through a SNIP address. This user account is created by the partition administrator and the user has access to resources, only within the partition.

Points to remember

Following are some points to remember when providing role-based access in a partition.

1. NetScaler users accessing the GUI through NSIP address will use default partition authentication configuration to log on to the appliance.
2. Partition system users accessing the GUI through partition SNIP address will use partition specific authentication configuration to log on to the appliance.
3. Partition user created in a partition cannot login using NSIP address.
4. NetScaler user bound to a partition cannot login using partition SNIP address.
5. External users accessing a partition through external server configuration as LDAP, Radius, or TACACS added in the partition. The user must access using SNIP address to directly log onto the partition.

Use case for managing role Based access in an partitioned setup

Consider a scenario where an enterprise organization, www.example.com has multiple business units and a centralized administrator who manages all instances in their network. However, they want to provide exclusive user privileges and environment for each business unit.

Following are the administrators and users managed by default partition authentication configuration and partition specific configuration in a partitioned appliance.

John: Root Administrator

George: Partition Administrator

Adam: System User

Jane: Partition User

John, is the root administrator of a partitioned NetScaler appliance. John manages all user accounts and administrative user accounts across partitions (for example, P1, P2, P3, P4, and P5) within the appliance. He provides granular role-based access to entities from the default partition of the appliance. John creates user accounts and assigns partition access to each account. George being a network engineer within the organization prefers to have a role based access to few applications running on partition P2. Based on user management, John creates a partition administrator role for George and associates his user account with partition-admin command policy in P2 partition. Adam being another network engineer prefers to access an application running on P2. John creates a system user account for Adam and associates his user account to P2 partition. Once his account is created, Adam can log into the appliance to access the NetScaler Management interface through NSIP address and can switch to partition P2 based on user/group binding.

Suppose, Jane who is another network engineer wants to directly access an application running only on partition P2, George (partition administrator) can create a partition user account for her and associate her account with command policies for authorization privileges. Jane's user account created within the partition is now directly associated with P2. Now Jane can access the NetScaler Management interface through SNIP address and cannot switch to any other partition.

> **Note**
>
> If Jane's user account is created by a partition administrator in partition P2, she can access the NetScaler Management interface only through SNIP address (created within the partition) and not permitted to access the interface through NSIP address. Similarly, if Adam's user account is created by a root administrator in the default partition and is bound to P2 partition, he can access the NetScaler Management interface only through NSIP address or SNIP address created in the default partition (with management access enabled) and not permitted to access the partition interface through SNIP address created in the administrative partition.

Configure roles and responsibilities for partition administrators

Following are the configurations performed by a root administrator in a default partition.

Creating administrative partitions and system users – A root administrator creates administrative partitions and system users in the default partition of the appliance. The administrator then associates the users to different partitions. If you are bound to one or more partitions, you can switch from one partition to another based on user bindings. Also, your access to one or more bound partitions is authorized only by the root administrator.

Authorizing system user as partition administrator for a specific partition – Once a user account is created, the root administrator switches to a specific partition and authorizes the user as the partition administrator. This is done by assigning partition-admin command policy to the user account. Now, the user can access the partition as partition administrator and manage entities within the partition.

Following are the configurations perform by a partition administrator in an administrative partition.

Configuring SNIP address in an administrative partition- The partition administrator logs on to the partition and creates a SNIP address and provides management access to the address.

Creating and Binding a Partition System User with Partition Command Policy -The partition administrator creates partition users and defines the scope of user access. This is done by binding the user account to partition command policies.

Creating and Binding a Partition System User Groups with Partition Command Policy -The partition administrator creates partition user groups and defines the scope of user group access. This is done by binding the user group account to partition command policies.

Configuring External Server authentication for external users (optional)-This configuration is done for authenticating external TACACS users accessing the partition using SNIP address.

Following are the tasks performed in configuring role-based access for partition users in an Administrative Partition.

1. Creating an Administrative Partition – Before you create partition users in an administrative partition, you must first create the partition. As a root administrator, you can create a partition from the default partition using the configuration utility or a command line interface.
2. Switching user access from default partition to partition P2 – If you are partition administrator accessing the appliance from the default partition, you can switch from default partition to a specific partition (for example, partition P2) based on user binding.
3. Adding SNIP address to the Partition user account with Management access enabled-Once you have switched your access to an administration partition, you must create a SNIP address and provide management access to the address.
4. Creating and Binding a Partition System User with Partition Command Policy-If you are a partition administrator, you can create partition users and define the scope of user access. This is done by binding the user account to partition command policies.
5. Creating and Binding Partition user group with Partition Command Policy-If you are a partition administrator, you can create partition user groups and define the scope of user access control. This is done by bind the user group account to partition command policies.

Configuring External Server authentication for external users (optional)-This configuration is done for authenticating external TACACS users accessing the partition using SNIP address.

Benefits of using admin partitions

You can avail the following benefits by using admin partitions for your deployment:

- Allows delegation of administrative ownership of an application to the customer.
- Reduces the cost of ADC ownership without compromising on performance and ease-of-use.
- Safeguards from unwarranted configuration changes. In a non-partitioned NetScaler appliance, authorized users of other application could intentionally or unintentionally change

configurations that are required for your application. This could lead to undesirable behavior. This possibility is reduced in a partitioned NetScaler appliance.

- Isolates traffic between different applications by the use of dedicated VLANs for each partition.
- Accelerates and allows to scale application deployments.
- Allows application-level or localized management and reporting.

Let us analyze a couple of cases to understand the scenarios in which you can use admin partitions.

User case 1: How Admin partition is used in an enterprise network

Let us consider a scenario faced by a company named **Foo.com**.

- **Foo.com** has a single NetScaler.
- There are five departments and each department has one application that requires to be deployed with the NetScaler.
- Each application must be managed independently by a different set of users or administrators.
- Other users must be restricted from accessing the configurations.
- The application or back-end must be able to share resources like IP addresses.
- The global IT department must be able to control NetScaler-level settings which must be common to all partitions.
- Applications must be independent of one another. An error in configuration of one application must not affect the other.

A non-partitioned NetScaler would not be able to satisfy these requirements. However, you can achieve all these requirements by partitioning a NetScaler.

Simply create a partition for each of the applications, assign the required users to the partitions, specify a VLAN for each partition, and define global settings on the default partition.

Use case 2: How an admin partition is used by a service provider

Let us consider a scenario faced by a service provider named **BigProvider**:

- BigProvider has 5 customers: 3 small enterprises and 2 large enterprises.
- **SmallBiz**, **SmallerBiz**, and **StartupBiz** need only the most basic NetScaler functionality.
- **BigBiz** and **LargeBiz** are larger enterprises and have applications that attract a lot of traffic. They would like to use some of the more complex NetScaler functionality.

In a non-partitioned approach, the NetScaler administrator would typically use a NetScaler SDX appliance and provision a NetScaler instance for each customer.

This solution suits **BigBiz** and **LargeBiz** because their applications need the undiminished power of the entire non-partitioned NetScaler appliance. However, this solution might not be as cost effective for servicing **SmallBiz**, **SmallerBiz**, and **StartupBiz**.

Therefore, **BigProvider** decides on the following solution:

- Using a NetScaler SDX appliance to bring up dedicated NetScaler instances for **BigBiz** and **LargeBiz**.
- Using a single NetScaler which is partitioned into three partitions, one each for **SmallBiz**, **SmallerBiz**, and **StartupBiz**.

The NetScaler administrator (superuser) creates an admin partition for each of these customers, specifies the users for the partitions, specifies the NetScaler resources for the partitions, and specifies the VLAN to be used by the traffic that is destined for each of the partitions.

Supported NetScaler configurations

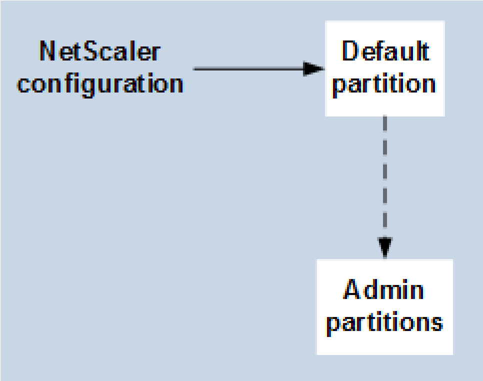
November 10, 2020

Depending on the NetScaler configuration and the partition in which the configuration is performed, NetScaler configurations can be categorized into the following three types of configurations.

> **Note**
>
> - Admin partitions cannot be set up on a NetScaler cluster. This means that a NetScaler cluster cannot be partitioned.
> - Admin partitions cannot be set up on a NetScaler MPX-FIPS appliance.
> - Case 3 lists the NetScaler features that are not supported in admin partitions.
> - Load balancing templates are not supported in admin partitions.

Case 1 (global configurations)

Configurations that can be performed ONLY in the default partition and which are available or impact all the admin partitions.

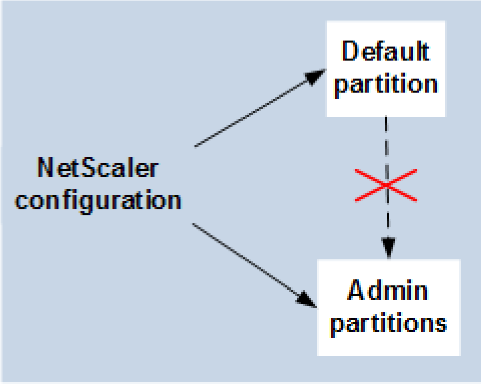


- Updates to built-in entities for monitors, TCP profiles, HTTP profiles, and so on.

- Updates to global parameters for syslog, nslog, weblog, content switching, IPSEC, SIP, DHCP, Surge protection, TCP buffering, and system collection.
- High availability (HA) configurations
- Interface and VLAN changes
- User configurations

Case 2 (partition-specific configurations)

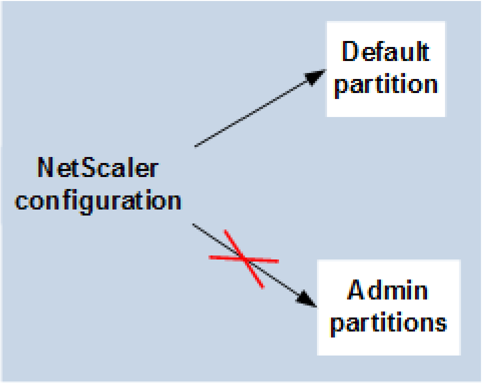
Configurations that can be performed independently in default and admin partitions. These configurations are applicable only to the partition in which they are performed.



- Getting traffic level statistics for a partition.
- Partition admin can update IP bindings for VLAN which is bound to that partition. But cannot update the interface bindings.
- Clearing NetScaler configurations.
- Feature-specific parameters for the following features: AppFlow, AppQoE, HTTP compression, DNS, TCP, HTTP, encryption, responder, rewrite, and SSL.
- Feature-specific configurations such as virtual servers, services, and monitors.

Case 3

Configu rations that cannot be performed on admin partitions. These features can be configured in the default partition, but there is no impact on admin partitions.



> **Note**
>
> Configurations that are supported on admin partitions for a particular release are marked as **Yes**.

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Networking	Traffic Domain	No (Not supported from build 60.13 onwards)	No	No	No
Policy	Extensibility	Yes	Yes	Yes	Yes
Load Balancing	DBS AutoScale	Yes	Yes	Yes	Yes
Load Balancing	DNSSEC	No	No	Yes	Yes
Load Balancing	Diameter	Yes	Yes	Yes	Yes
Load Balancing	RTSP	No	No	No	No
Load Balancing	Sure Connect	Yes	Yes	Yes	Yes
Load Balancing	Autoscale Service Group	Yes	Yes	Yes	Yes
Manageability	RBA External Authentication	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Manageability	RISE Cisco	No	No	No	No
Manageability	ACI-Cisco	Yes	Yes	Yes	Yes
Manageability	AppExpert	Yes	Yes	Yes	Yes
Manageability	HDX Insight	No	No	No	No
Manageability	Insight	No	No	No	No
VPN	Cloudbridge Connector	No	No	No	No
VPN	Citrix Gateway or SSL VPN	No	No	No	No
VPN	SSL VPN ICA Proxy	No	No	No	No
VPN	Web Interface on Citrix ADC	No	No	No	No
SSL	SSL Profile	Yes	Yes	Yes	Yes
SSL	SSL-FIPS	No	No	No	No
SSL	External-HSM	No	No	No	No
Infra	Cache Redirection	No	No	No	No
Infra	Integrated Caching (Restricted Feature)	Yes	Yes	Yes	Yes
Network	VXLAN	Yes	Yes	Yes	Yes
Network	Graceful Shutdown	Yes	Yes	Yes	Yes
Network	LSN	No	No	No	No
Network	IPv6 Ready Logo	Yes	Yes	Yes	Yes
Network	Vpath	Yes	Yes	Yes	Yes
Load Balancing	Datastream	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
Logging	Web logging	Yes	Yes	Yes	Yes
Network	L2 Param/L3 Param	Yes	Yes	Yes	Yes
Network	GRE Tunnel	Yes	Yes	Yes	Yes
Loading Balancing	Scriptable Monitoring	Yes	Yes	Yes	Yes
Load Balancing	GSLB	Yes	Yes	Yes	Yes
Infra	Connection Mirroring	Yes	Yes	Yes	Yes
Infra	FEO	Yes	Yes	Yes	Yes
Infra	Nstrace	Yes	Yes	Yes	Yes
Load Balancing	Priority Queuing	Yes	Yes	Yes	Yes
Network	HDOSP	Yes	Yes	Yes	Yes
Network	Netprofile (Restricted Feature)	Yes	Yes	Yes	Yes
Network	Networking (Restricted Feature)	Yes	Yes	Yes	Yes
Network	VRRP (Restricted Feature)	Yes	Yes	Yes	Yes
Logging	Audit Logging (SYSLOG-TCP, LB of syslog servers, SNIP support, and FQDN support for syslog)	Yes	Yes	Yes	Yes

Feature Component	Citrix ADC Feature	NetScaler 11.1	NetScaler 12.0	Citrix ADC 12.1	Citrix ADC 13.0
VPN	Citrix Gateway	No	No	No	No
VPN	AAA-TM	Yes	Yes	Yes	Yes
AppFlow	AppFlow	No	Yes (Ipfix only)	Yes (Ipfix only)	Yes
AppFw	Application Firewall	No	No	No	No
URL transformation	URL transformation	No	No	No	No
Load Balancing	TCP Buffering (Restricted Feature)	No	No	No	No
Policies	OCSP Responder	Yes	Yes	Yes	Yes
Audit Log	SYSLOG-TCP	No	Yes	Yes	Yes
Optimization	Front-end-optimization	No	Yes	Yes	Yes
AppQoE	AppQoE	Yes	Yes	Yes	Yes

Configure admin partitions

September 21, 2020

> **Important**
>
> - Only superusers are authorized to create and configure admin partitions.
> - Unless specified otherwise, configurations to set up an admin partition must be done from the default partition.

By partitioning a NetScaler appliance, you are in-effect creating multiple instances of a single NetScaler appliance. Each instance has its own configurations and the traffic of each of these partitions is isolated from the other by assigning each partition a dedicated VLAN or a shared VLAN.

A partitioned NetScaler has one default partition and the admin partitions that are created. To set up an admin partition, you must first create a partition with the relevant resources (memory, maximum bandwidth, and connections). Then, specify the users that can access the partition and the level of authorization for each of the users on the partition.

Accessing a partitioned NetScaler is the same as accessing a non-partitioned NetScaler: through the NSIP address or any other management IP address. As a user, after you provide your valid logon credentials, you are taken to the partition to which you are bound. Any configurations that you create are saved to that partition. If you are associated with more than one partition, you are taken to the first partition with which you were associated. If you want to configure entities on one of your other partitions, you must explicitly switch to that partition.

After accessing the appropriate partition, configurations that you perform are saved to that partition and are specific to that partition.

> **Note**
>
> - NetScaler superusers and other non-partition users are taken to the default partition.
> - Users of all the 512 partitions can log in simultaneously.

Tip

To access a partitioned NetScaler appliance over HTTPS by using the SNIP (with management access enabled), make sure that each partition has the certificate of its partition administrator. Within the partition, the partition admin must do the following:

1. Add the certificate to the NetScaler.

   add ssl certKey ns-server-certificate -**cert** ns-server.cert -**key** ns-server.key

2. Bind it to a service named "nskrpcs-<SNIP>-3009", where <SNIP> must be replaced with the SNIP address, in this case 100.10.10.1.

   bind ssl service nskrpcs-100.10.10.1-3009 -**certkeyName** ns-server-certificate

Partition resource limiting

In a partitioned NetScaler appliance, a network administrator can create a partition with partition resources such as memory, bandwidth, and connection limit configured as unlimited. This is done by specifying Zero as the partition resource value, where Zero indicates the resource is unlimited on the partition and it can be consumed up to system limits. Partition resource configuration is useful when you migrate a traffic domain deployment to an administrative partition or if you do not know about resource allocation limit for a partition in a given deployment.

Resource limit for an administrative partition is as follows:

1. **Partition memory**. This is the maximum allocated memory for a partition. You must make sure to specify the values when creating a partition.

> **Note**
>
> From NetScaler 12.0 onwards, when you create a partition, you can set the memory limit to Zero. If a partition is already created with a specific memory limit, you can reduce the limit to any value or set the limit as Zero.

Parameter: maxMemLimit

Maximum memory is allocated in megabytes in a partition. A zero value indicates the memory is unlimited on the partition and it can consume up to the system limits.

Default value: 10

2. **Partition bandwidth**. Maximum allocated bandwidth for a partition. If you specify a limit, make sure it is within the appliance's licensed throughput. Otherwise, you are not limiting the bandwidth that can be used by the partition. The specified limit is accountable for the bandwidth that the application requires. If the application bandwidth exceeds the specified limit, packets are dropped.

   > **Note**
   >
   > From NetScaler 12.0 onwards, when you can create a partition, you can set the partition bandwidth limit to Zero. If a partition is already created with a specific bandwidth, you can reduce the bandwidth or set the limit as Zero.

   Parameter: maxBandwidth

   Maximum bandwidth is allocated in Kbps in a partition. A zero value indicates the bandwidth is unrestricted. That is, the partition can consume up to the system limits.

   Default value: 10240

   Maximum Value: 4294967295

3. **Partition connection**. Maximum number of concurrent connections that can be open in a partition. The value must accommodate the maximum simultaneous flow expected within the partition. The partition connections are accounted from the partition quota memory. Previously, the connections were accounted from the default partition quota memory. It is configured only on the client-side, not on the back-end server-side TCP connections. New connections cannot be established beyond this configured value.

   > **Note**
   >
   > From NetScaler 12.0 onwards, you can create a partition with number of open connections set to Zero. If you have already created a partition with a specific number of open connections, you can reduce the connection limit or set the limit as Zero.

   Parameter: maxConnections

Maximum number of concurrent connections that can be open in the partition. A zero value indicates no limit on number of open connections.

Default value: 1024

Minimum value: 0

Maximum Value: 4294967295

Configure an admin partition

To configure an admin partition, complete the following tasks.

Access an admin partition

To access in an admin partition by using the command line interface

1. Log on to the NetScaler appliance.
2. Check if you are in the correct partition. The command prompt displays the name of the currently selected partition.
3. If yes, skip to the next step.
4. If no, get a list of the partitions with which you are associated and switch over to the appropriate partition.
    - **show system user** <username>
    - **switch ns partition** <partitionName>
5. Now, you can perform the required configurations just as a non-partitioned NetScaler.

To access an admin partition by using the configuration utility

1. Log on to the NetScaler appliance.
2. Check if you are in the correct partition. The top bar of the graphical user interface displays the name of the currently selected partition.
    - If yes, skip to the next step.
    - If no, navigate to **Configuration > System > Administrative Partitions > Partitions**, right-click the partition to which you want to switch, and select **Switch**.
3. Now, you can perform the required configurations just as a non-partitioned NetScaler.

Add an admin partition

The root administrator adds an administrative partition from the default partition and binds the partition with VLAN 2.

To create an administrative partition by using the command line interface

At the command prompt, type:

```
add partition <partitionname>
```

Switch user access from default partition to an admin partition

Now switch user access from default partition to partition Par1.

To switch a user account from default partition to an admin partition by using the command line interface:

At the command prompt, type:

```
Switch ns partition <pname>
```

Adding SNIP address to a partition user account with management access enabled

In the partition, create SNIP address with management access enabled.

To add SNIP address to the partition user account with management access enabled by using the command line interface:

At the command prompt, type:

```
> add ns ip <ip address> <subnet mask> -mgmtAccess enabled
```

Create and Bind a partition user with partition command policy

In partition, create a partition system user and bind the user with partition-admin command policies.

To create and bind a partition system user with partition command policy by using the command line interface:

At the command prompt, type:

```
> add system user <username> <password>
Done
```

Creating and binding partition user group with partition command policy

In Partition Par1, create a partition system user group and bind the group with partition command policy such as partition admin, partition read-only, partition-operator, or partition-network.

To create and bind a partition user group with partition command policy by using the command line interface:

```
> add system group <groupName>
> bind system group <groupname> (-userName | -policyName <cmdpolicy> <
    priority> | -partitionName)
```

Configuring external server authentication for external users

In partition Par1 you can configure an external server authentication to authenticate external TACACS users accessing the partition through SNIP address.

To configure external server authentication for external users by using the command line interface:

At the command prompt, type:

```
add authentication tacacsaction <name> -serverip <IP> -tacacsSecret <
    secret key> -authorization ON -accounting ON
add authentication policy <policname> -rule true -action <name>
bind system global <policyname> -priority <value>1
```

Configure a Partition System User Accounts in a Partition by using the GUI

To configure a partition user account in an administrative partition, you must create a partition user or a partition user group and bind it partition command policies. Also, you can configure the external server authentication for an external user.

To create a partition user account in a partition by using the GUI

Navigate to **System > User Administration**, click **Users** to add a partition system user and bind the user to command policies (partitionadmin/partitionread-only/partition-operator/partition-network).

To create a partition user group account in a partition by using the GUI

Navigate to **System > User Administration**, click **Groups** to add a partition system user group and bind the user group to command policies (partitionadmin/partitionread-only/partition-operator/partition-network).

To configure External server authentication for external users by using the GUI

Navigate to **System > Authentication > Basic Actions** and click **TACACS** to configure TACACS server for authenticating external users accessing the partition.

Sample Configuration

The following configuration shows how to create a partition user or a partition user group and bind it partition command policies. Also, how to configure the external server authentication for authenticating an external user.

```
add partition Par1
switch ns partition Par1
> add ns ip 10.102.29.203 255.255.255.0 -mgmtAccessenabled
> add system user John Password
> bind system user Jane partition-read-only -priority 1
> add system group Retail
> bind system group Retail -policyname partition-network 1 (where 1 is
   the priority number)
> bind system group Retail  - username Jane
> add authentication tacacssaction tacuser  - serverip 10.102.29.200  -
   tacacsSecret Password  - authorization ON  - accounting ON
> add authentication policy polname  - rule true  - action tacacsAction
> bind system global polname  - priority 1
```

Command Policies for a Partition Users and Partition User Groups in Administrative Partition

Commands to authorize an user account inside administrative partition	Command policies available inside an administrative partition (built-in policies)	User account access type
add system user	Partition-admin	SNIP (with management access enabled)
add system group	Partition-network	SNIP (with management access enabled)
add authentication <action, policy>, bind system global <policy name>	Partition-read-only	SNIP (with management access enabled)
remove system user	Partition-admin	SNIP(with management access enabled)

Commands to authorize an user account inside administrative partition	Command policies available inside an administrative partition (built-in policies)	User account access type
remove system group	Partition-admin	SNIP (with management access enabled)
bind system cmdpolicy to system user; bind system cmdpolicy to system group	Partition-admin	SNIP (with management access enabled)

Configure an LACP ethernet channel on the default admin partition

With Link Aggregation Control Protocol (LACP), you can combine multiple ports into a single, high-speed link (also called a channel). An LACP-enabled appliance exchanges LACP Data Units (LACPDU) over the channel.

There are three LACP configuration modes that you can enable in the default partition of a NetScaler appliance:

1. Active. A port in active mode sends LACPDUs. Link aggregation is formed if the other end of the Ethernet link is in the LACP active or passive mode.
2. Passive. A port in passive mode sends LACPDUs only when it receives LACPDUs. The link aggregation is formed if the other end of the Ethernet link is in the LACP active mode.
3. Disable. Link aggregation is not formed.

> **Note**
>
> By default, the link aggregation is disabled in the default partition of the appliance.

LACP exchanges LACPDU between devices connected by an Ethernet link. These devices are typically referred as an actor or partner.

A LACPDU data unit contains the following parameters:

- LACP Mode. Active, passive or disable.
- LACP timeout. The waiting period before timing out the partner or actor. Possible values: Long and Short. Default: Long.
- Port Key. To distinguish between the different channel. When key is 1, LA/1 is created. When key is 2, LA/2 is created. Possible values: Integer from 1 through 8. 4 through 8 is for cluster CLAG.
- Port Priority. Minimum value: 1. Maximum value: 65535. Default: 32768.
- System Priority. Uses this priority along with system MAC to form the system ID to uniquely identify the system during LACP negotiation with the partner. Sets system priority from 1 and 65535. The default value is set to 32768.

- Interface. Supports 8 interfaces per channel on NetScaler 10.1 appliance and supports 16 interfaces per channel on NetScaler 10.5 and 11.0 appliances.

After exchanging LACPDUs, the actor and partner negotiate the settings and decide whether to add the ports to the aggregation.

Configure and verify LACP

To configure and verify LACP on a NetScaler appliance by using the command line

1. Enable LACP on each interface.

   At the command prompt, type:

   ```
   set interface <Interface_ID> -lacpMode PASSIVE -lacpKey 1<!--NeedCopy
   -->
   ```

   When you enable LACP on an interface, the channels are dynamically created. Additionally, when you enable LACP on an interface and set lacpKey to 1, the interface is automatically bound to channel LA/1.

   > **Note**
   >
   > When you bind an interface to a channel, the channel parameters take precedence over the interface parameters, so the interface parameters are ignored. If a channel was created dynamically by LACP, you cannot perform add, bind, unbind, or remove operations on the channel. A channel dynamically created by LACP is automatically deleted when you disable LACP on all interfaces of the channel.

2. Set the system priority.

   At the command prompt, type:

   ```
   set lacp -sysPriority <Positive_Integer><!--NeedCopy-->
   ```

3. Verify that LACP is working as expected.

   "'show interface

   ~~~
   ```show channel<!--NeedCopy-->
   ~~~

   ```
   show LACP<!--NeedCopy-->
   ```

   > **Note**
   >
   > In some versions of Cisco IOS, running the switchport trunk native vlan <VLAN_ID> command causes the Cisco switch to tag LACP PDUs. This causes the LACP channel between

> the Cisco switch and the NetScaler appliance to fail. However, this issue does not affect the static link aggregation channels configured in the above procedure.

VLAN configuration for admin partitions

March 17, 2021

VLANs can be bound to a partition as a "Dedicated" VLAN or a "Shared" VLAN. Based on your deployment, you can bind a VLAN to a partition to isolate its network traffic from other partitions.

Dedicated VLAN – A VLAN bound only to one partition with the "Sharing" option disabled and must be a tagged VLAN. For example, in a client-server deployment, for security reasons a system administrator creates a dedicated VLAN for each partition on the server side.

Shared VLAN – A VLAN bound (shared across) to multiple partitions with the "Sharing" option enabled. For example, in a client-server deployment, if the system administrator do not have control over the client side network, a VLAN is created and shared across multiple partitions.

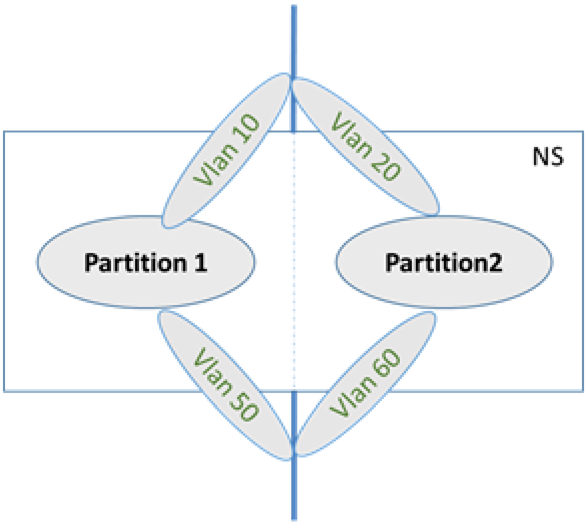
Shared VLAN can be used across multiple partitions. It is created in the default partition and you can bind a shared VLAN to multiple partitions. By default, a shared VLAN is bound to the default partition implicitly and hence it cannot be bound explicitly.

> **Important**
>
> - A NetScaler appliance deployed on any hypervisor (ESX, KVM, Xen, and Hyper-V) platform must comply with both the following conditions in a partition setup and traffic domain:
>   - Enable the promiscuous mode, MAC changes, MAC spoofing, or forged transmit for shared VLANs with partition.
>   - Enable the VLAN with `portgroup` properties of the virtual switch, if the traffic is through a dedicated VLAN.
> - Citrix recommends you to bind a Dedicated or Shared VLAN to multiple partitions. You can bind only a tagged VLAN to a partition. If there are untagged VLANs, you must enable them as "Shared" VLANs and then bind them to other partitions. It ensures that you control traffic packets (for example, LACP, LLDP, and xST P packets) handled in the default partition. If you have already bound an untagged VLAN for a partition in 11.0, see "Deployment procedure for upgrading a sharable VLAN to NetScaler 11.1 software" procedure.

In a partitioned (multitenant) NetScaler appliance, a system administrator can isolate the traffic flowing to a particular partition or partitions. It is done by binding one or more VLANs to each partition. A VLAN can be dedicated to one partition or shared across multiple partitions.

Dedicated VLANs

To isolate the traffic flowing into a partition, create a VLAN and associate it with the partition. The VLAN is then visible only to the associated partition, and the traffic flowing through the VLAN is classified and processed only in the associated partition.



To implement a dedicated VLAN for a particular partition, do the following.

1. Add a VLAN (V1).
2. Bind a network interface to VLAN as a tagged network interface.
3. Create a partition (P1).
4. Bind partition (P1) to the dedicated VLAN (V1).

Configure the following by using the CLI

- Create a VLAN

  `add vlan <id>`

Example

```
add vlan 100
```

- Bind a VLAN

  `bind vlan <id> -ifnum <interface> -tagged`

Example

```
bind vlan 100 - ifnum 1/8 -tagged
```

- Create a partition

`Add ns partition <partition name> [-maxBandwidth <positive_integer>][-maxConn <positive_integer>] [-maxMemLimit <positive_integer>]`

Example

```
Add ns partition P1 - maxBandwidth 200 - maxconn 50 - maxmemlimit
   90

Done
```

- Bind a partition to a VLAN

  `bind partition <partition-id> -vlan <id>`

Example

```
bind partition P1 - vlan 100
```

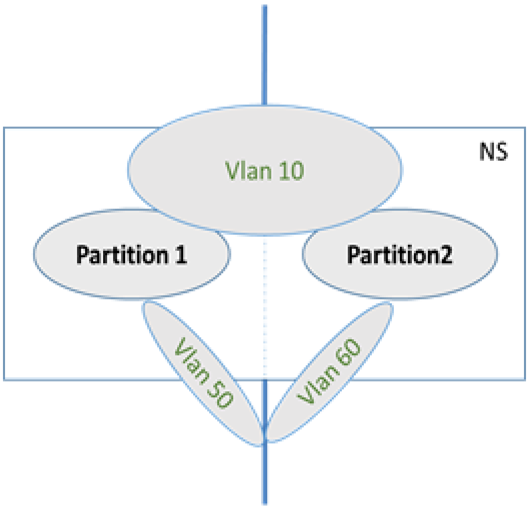
Configure a dedicated VLAN by using the NetScaler GUI

1. Navigate to **Configuration > System > Network > VLANs*** and click **Add** to create a VLAN.
2. On the **Create VLAN** page, set the following parameters:
   - VLAN ID
   - Alias Name
   - Maximum Transmission Unit
   - Dynamic Routing
   - IPv6 Dynamic Routing
   - Partitions Sharing
3. In the **Interface Bindings** section, select one or more interfaces and bind it to the VLAN.
4. In the **IP Bindings** section, select one or more IP addresses and bind to the VLAN.
5. Click **OK** and **Done**.

Shared VLAN

In a shared VLAN configuration, each partition has a MAC address, and traffic received on the shared VLAN is classified by MAC address. Only a Layer3 VLAN is recommended because it can restrict the subnet traffic. A partition MAC address is applicable and important only for a shared VLAN deployment.

The following diagram shows how a VLAN (VLAN 10) is shared across two partitions.



To deploy a shared VLAN configuration, do the following:

1. Create a VLAN with the sharing option 'enabled', or enable the sharing option on an existing VLAN. By default, the option is 'disabled'.
2. Bind partition interface to shared VLAN.
3. Create the partitions, each with its own PartitionMAC address.
4. Bind the partitions to the shared VLAN.

Configure a shared VLAN by using the CLI

At the command prompt, type one of the following commands to add VLAN or set the sharing parameter of an existing VLAN:

```
add vlan <id> [-sharing (ENABLED | DISABLED)]

set vlan <id> [-sharing (ENABLED | DISABLED)]

add vlan 100  - sharing ENABLED

set vlan 100  - sharing ENABLED
```

Bind a partition to a Shared VLAN by using the CLI

At the command prompt, type:

```
bind partition <partition-id> -vlan <id>

bind partition P1  - vlan 100

```

```
add ns partition P1  - maxBandwidth 200  - maxconn 50    - maxmemlimit 90
    -partitionMAC<mac_addr

Done
```

Configure a Partition MAC Address by using the CLI

```
set ns partition <partition name> [-partitionMAC<mac_addr>]

set ns partition P1  - partitionMAC 22:33:44:55:66:77
```

Bind partitions to a shared VLAN by using the CLI

```
bind partition <partition-id> -vlan <id>

bind partition <partition-id> -vlan <id>

bind partition P1  - vlan 100

bind partition P2  - vlan 100

bind partition P3  - vlan 100

bind partition P4  - vlan 100
```

Configure Shared VLAN by using the NetScaler GUI

1. Navigate to **Configuration > System > Network > VLANs** and then select a **VLAN** profile and click **Edit** to set the partition sharing parameter.
2. On the **Create VLAN** page, select the **Partitions Sharing** check box.
3. Click **OK** and then **Done**.

Shared VLAN in a partitioned deployment on a NetScaler SDX appliance

Log on to a Storage Virtualization Manager (SVM) appliance and assign each partition's MAC (VMAC) to a NetScaler VPX appliance.

- **Maximum memory limit.** Must be configured as the memory that is required for each admin partition. Make sure that you set the appropriate
  value when creating the partition.

Once an admin partition is created, the memory limit cannot be decreased. The memory limit can however be increased when required or more specifically, when there is execution failure due to insufficient memory in a partition. It is done provided sufficient memory is available in the default partition.

> **Note**
>
> From NetScaler 11.0 build 64.x onwards, you can set the memory limit to a minimal value of 5 MB, when creating the admin partition. It can be useful for lighter deployments of the NetScaler appliance.

- **Maximum bandwidth**. The maximum bandwidth that is used by an admin partition. This value must be limited to the appliance's licensed throughput.
  Otherwise, in effect, you are NOT limiting the bandwidth that is used by the admin partition.

  It must be configured such that it accounts for the bandwidth that the application requires. If the application bandwidth exceeds the configured value, packets are dropped. It accounts for incoming and outgoing packets. The maximum bandwidth can be increased or decreased when required.

> **Note**
>
> - The default value is 10240 kbps, minimum value is 0, and maximum value is 4294967295 kbps.
> - Setting this parameter to its minimum value (0) means that you are not assigning any bandwidth to the partition. Traffic received for this partition is dropped.
> - It is not the guaranteed bandwidth available for the admin partition. After a partition is configured with a maximum bandwidth value, the actual bandwidth assigned depends on the appliance's licensed throughput.

- **Maximum number of connections**. Must be configured such that it accounts for the maximum simultaneous flows expected within a partition. It is configured only on the client-side and not on the back-end server-side TCP connections. New connections cannot be established beyond this configured value.

  The maximum number of connections can be increased or decreased when required.

> **Note**
>
> - When the bandwidth and number of connections crosses the threshold value, if SNMP is configured, traps are sent with the relevant information.
> - After creating a partition, inform the users that the NetScaler configurations they perform will be isolated from users who are not members of the partition.

> - Make sure the relevant users, command policies, VLANs, and bridge groups are available on the NetScaler appliance.
> - For deployments that have a large size of NetScaler configuration and large quantum of traffic, Citrix recommends you to increase the default values for the maximum memory limit, maximum bandwidth, and maximum number of connections.
> - Shared VLAN in a partitioned appliance does not support the dynamic routing protocol.

Configure shared VLAN by using the NetScaler CLI

- Create a partition and configure the NetScaler resources for that partition.

  `add ns partition <partitionName> [-maxBandwidth <positive_integer>] [-maxConn <positive_integer>] [-maxMemLimit <positive_integer>]`

> **Note**
>
> Check the rate limiting content provided for tips to update the maximum memory limit, maximum
> bandwidth, and maximum number of connections.

- Associate the appropriate users with the partition.

  `bind system user <name> -partitionName <string>`

- Specify the level of authorization for each user by associating one of the following command policies: *partition operator, partition-read-only, partition-network, and partition-admin.*

  `bind system user <name> <policyName> <priority>`

- Configure the VLAN through which traffic for this partition must be routed. You can use bridge groups instead of VLANs to route the traffic.
  - Add the VLAN and bind the required interfaces to it.

    ```
    - add vlan <id>
    - bind vlan <id> if num <interface>
    ```

    > **Note**
    >
    > When a VLAN is bound to an admin partition, its IP address bindings are lost. To make sure that the VLAN continues to have the IP address, create the IP address on the admin partition and then bind it to that VLAN.

  OR

  - Add the bridge group and bind the required VLANs to it.

```
-  add bridgegroup <id>
-  bind bridgegroup <id> -vlan <id>
```

- Bind the VLAN or bridge group to the partition.

  ```
  bind ns partition <partitionName> -vlan <positive_integer>
  ```

  OR

  ```
  bind ns partition <partitionName> -bridgegroup <positive_integer>
  ```

  > **Note**
  >
  > Use the show VLAN or the show bridge group command to view the partitions associated with that VLAN or bridge group.

- Verify the configurations of the partition.

  ```
  show ns partition <partitionName>
  ```

  > **Note**
  >
  > You can also use the `stat ns partition` command to view partition configurations.

- Save the configuration.

  ```
  save ns config
  ```

Configure Shared VLAN by using the NetScaler GUI

- Navigate to **System > Partition Administration**, click **Add**, and do the following:
  - Create and configure the resources for the admin partition.
  - Specify the VLANs or bridge groups to associate with partition.
  - Associate users with the partition.

    > **Note**
    >
    > Make sure you bind users who are not yet associated with partition type command policies.

- Navigate to **System > User Administration**, and to the partition user, bind the appropriate command policy. The command policy must be one of the partition-entries. The choice depends on the level of authorization you intend the user to have.
- Save the configuration.

SNMP support for admin partitions

September 21, 2020

A partitioned NetScaler appliance uses SNMP infrastructure for partition rate limiting and for monitoring partition resource utilization details.

SNMP traps for admin partition rate limiting

On a partitioned NetScaler appliance, a PARTITION-RATE-LIMIT alarm can generate nine SNMP traps for notifying a partition resource (such as bandwidth, connection, or memory) has reached its limit or returned to normal.

The following nine SNMP traps are generated when:

- **partitionCONNThresholdReached**. The number of active connections for a partition exceeds its high threshold percentage.
- **partitionCONNThresholdNormal**. The number of active connections is less than or equal to the normal threshold percentage.
- **partitionBWThresholdReached**. The partition's bandwidth usage reaches its high threshold percentage.
- **partitionMEMThresholdReached**. The current memory usage of the partition exceeds its high threshold percentage.
- **partitionMEMThresholdNormal**. The current memory usage of the partition becomes less than or equal to the normal threshold percentage.
- **partitionMEMLimitExceeded.** The current memory usage of the partition exceeds its memory limit percentage.
- **partitionCONNLimitExceeded**. The number of active connections for a partition exceeds its configured limit and new connections are being dropped.
- **partitionCONNLimitNormal**. The number of active connections for a partition goes below its configured limit and the partition can now accept a new connection.
- **partitionBWLimitExceeded**. The current bandwidth usage for a partition has exceeded its configured limit.

The threshold values for the SNMP traps are non-configurable and are as follows are:

- High threshold = 80% (applicable for all partition rate limit traps)
- Low threshold = 60 % (applicable for all partition rate limit traps)
- Memory limit = 95% (applicable only for partition memory traps)

Configuring PARTITION-RATE-LIMIT alarm

To configure PARTITION-RATE-LIMIT alarm in a specific partition and enable generation of the SNMP trap messages.

1. Enable PARTITION-RATE-LIMIT Alarm
2. Configure PARTITION-RATE-LIMIT Alarm
3. Configure SNMP Trap Destination

To enable PARTITION-RATE-LIMIT alarm by using the NetScaler CLI

At the command prompt, type the following commands:

```
enable snmp alarm PARTITION-RATE-LIMIT

show snmp alarm PARTITION-RATE-LIMIT
```

To configure PARTITION-RATE-LIMIT Alarm by using the NetScaler CLI

At the command prompt, type the following command

```
set snmp alarm PARTITION-RATE-LIMIT [-state ( ENABLED | DISABLED )] [-
    severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure SNMP trap destination by using the NetScaler CLI

At the command prompt, type the following command:

```
add snmp trap <trapClass> <trapDestination> [-version <version>] [-td <
    positive_integer>] [-destPort <port>] [-communityName <string>] [-
    srcIP <ip_addr|ipv6_addr>] [-severity <severity>] [-allPartitions (
     ENABLED | DISABLED )]
```

To configure PARTITION-Rate-Limit alarm by using the NetScaler GUI

Navigate to **System > SNMP > Alarms**, select **PARTITION-RATE-LIMIT** alarm and configure the alarm parameters.

To configure SNMP trap destination by using the NetScaler GUI

Navigate to **System > SNMP > Trap**, specify the IP address of the destination device.

SNMP monitoring for partition resource utilization

Using SNMP, you can monitor a partition's resource (such as bandwidth, connection, and memory) utilization details at real time on a NetScaler appliance. This is done by sending a SNMP request (such as SNMP GET, SNMP GET BULK, SNMP GETNEXT, or SNMP WALK) from the SNMP Manager.

> **Note**
>
> In order to monitor partition resources, you must configure the SNMP community in the default partition. Wherein, the *partitionTable* is maintained in the default partition, and the SNMP communication is done through the NSIP address of the appliance.

Consider a scenario where a NetScaler administrator wants to know the bandwidth usage of partition P1 on the appliance. The SNMP Manager retrieves this information by sending an SNMP GET request on the corresponding OID (partitionCurrentBandwidth) to the NSIP address of the appliance. The SNMP agent on the default partition retrieves and sends the current bandwidth usage of P1 to the SNMP Manager through the NSIP address.

The following table lists the SNMP counters which are part of *partitionTable* and its description:

SNMP Parameter	SNMP OID	Description
partitionName	1.3.6.1.4.1.5951.4.1.1.88.1.1	Partition name
partitionCurrentBandwidth	1.3.6.1.4.1.5951.4.1.1.88.1.2	Current bandwidth usage of the partition.
partitionCurrentConnections	1.3.6.1.4.1.5951.4.1.1.88.1.3	Current number of active connections of the partition.
partitionMemoryUsagePcnt	1.3.6.1.4.1.5951.4.1.1.88.1.4	Current Memory usage (in percentage) of the partition.

AppExpert

September 21, 2020

The following topics provide a conceptual reference and configuration instructions for the AppExpert and other features of the NetScaler appliance.

> **Note**
>
> For information about policy extensions, see Policy Extensions.

- Action Analytics: Collects run-time statistics on the basis of pre-defined criteria. When used with policies, the feature also provides you with the infrastructure for automatic, real-time traffic optimization.

- AppExpert Applications and Templates: Simplify configuration steps for the Citrix® NetScaler® appliance by using applications, application templates, Citrix Gateway applications, and entity templates.
- AppQoE: Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the NetScaler appliance into a single integrated feature that takes advantage of a new queuing mechanism, fair queuing.
- Entity Template: Describes how to use entity templates to set up and configure individual NetScaler entities, such as a policy or virtual server. An entity template provides a specification and a set of defaults for the object.
- HTTP Callouts: An HTTP request that the NetScaler appliance generates and sends to an external application when certain criteria are met during policy evaluation.
- Pattern Sets: Allow string matching during the evaluation of a default syntax policy.
- Policies and Expressions: Rules that determine the operations that the NetScaler appliance must perform.
- Rate Limiting: Defines the maximum load for a given network entity or virtual entity on the NetScaler appliance.
- Responder: Bases responses on who sends the request, where it is sent from, and other criteria with security and system management implications.
- Rewrite: Rewrites information in the requests or responses handled by the NetScaler appliance.
- String Maps: Perform pattern matching in all NetScaler features that use the default policy syntax.

Action analytics

September 21, 2020

The performance of your website or application depends on how well you optimize the delivery of the most frequently requested content. Techniques such as caching and compression help accelerate the delivery of services to clients, but you need to be able to identify the resources that are requested most frequently, and then cache or compress those resources. You can identify the most frequently used resources by aggregating real-time statistics about website or application traffic. Statistics such as how frequently a resource is accessed relative to other resources and how much bandwidth is consumed by those resources help you determine whether those resources need to be cached or compressed to improve server performance and network utilization. Statistics such as response times and the number of concurrent connections to the application help you determine whether you must enhance server-side resources.

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

When configuring the action analytics feature, you specify the request attributes for which you want to collect statistical data, for example, URLs and HTTP methods by configuring default syntax expressions in an entity called a selector. Then, you configure an identifier to configure settings such as the sampling interval and sample count. You also configure a policy that enables the appliance to evaluate traffic as specified by the selector-identifier pair. Finally, you bind the policy to a bind point to begin collecting statistics.

The appliance also provides you with a set of built-in selectors, identifiers, and responder policies that you can use to get started with the feature.

The appliance aggregates the following statistics:

- The number of requests.
- The bandwidth consumed by the requests.
- The response time.
- The number of concurrent connections.

You can configure the feature to perform run-time sorting of the records on an attribute of your choice. You can view the statistical data by using either the command-line interface or the Stream Sessions tool in the GUI.

Configure a selector

September 21, 2020

A selector is a filter for identifying requests. It consists of up to five individual default syntax expressions that identify request attributes such as the client IP address and the URL in the request. Each expression is a non-compound default syntax expression and is considered to be in an AND relationship with the other expressions. Following are some examples of selector expressions:

- `HTTP.REQ.URL`
- `CLIENT.IP.SRC`
- `HTTP.RES.BODY(1000).AFTER_STR("<string>").BEFORE_STR("<string>")"`
- `CLIENT.IP.SRC.SUBNET(24)`

Selectors are used in rate limiting and action analytics configurations. A selector is optional in a rate limiting configuration but is required in an action analytics configuration.

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

If you modify an expression in a selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a selector named myLimitSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myLimitSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

The NetScaler appliance provides built-in selectors for some of the most common use cases. Refer to the pdf.

You can also configure a selector with expressions that identify the request attributes of your choice. For example, you might want to create a record for a request that arrives with a specific header. To evaluate the header, you can add HTTP.REQ.HEADER("") to the selector that you intend to use.

To configure a selector by using the command line interface:

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

- `add stream selector <name> <rule> ...`
- `show stream selector`

Example

```
> add stream selector myselector HTTP.REQ.URL CLIENT.IP.SRC
Done
> show stream selector myselector
    Name: myselector
    Expressions:
        1) HTTP.REQ.URL
        2) CLIENT.IP.SRC
Done
```

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

To modify or remove a selector by using the command line interface:

- To modify a selector, type the set stream selector command, the name of the selector, and the rule parameter with the expressions. Enter the existing expressions that you want to retain, along with the new expressions that you want to add.
- To remove a selector, type the rm stream selector command and the name of the selector.

To configure a selector by using the GUI:

1. Navigate to **AppExpert** > **Action Analytics** > **Selectors**.
2. In the details pane, do one of the following:
   - To create a selector, click **Add**.
   - To modify a selector, select the selector, and then click **Edit**.
3. In the **Create Selector** or **Configure Selector** page, set the following parameters:
   - Name
   - Expressions
     To add the expression to the selector configuration, click
     Add. To remove an expression from the selector configuration, in the
     Expression box, select the expression, and then click
     Delete.
     Note: In the
     Expressions box, enter a valid parameter. For example, enter HTTP. Then, enter a period after this parameter. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. To select the next keyword in this expression prefix, double-click the selection in the drop-down menu. The Expressions text box displays both the first and second keywords for the expression prefix, for example,
     HTTP.REQ. Continue adding expression components until the complete expression is formed.
4. Click **Add**.
5. Continue adding up to five non-compound expressions.
6. Click **OK** and then **Close**.

Configure a stream identifier

September 21, 2020

You configure a stream identifier to specify parameters for collecting statistical data from requests identified by a given selector. An identifier specifies the selector to be used, the statistics collection interval, the sample count, and the field on which the records are to be sorted.

The NetScaler appliance includes the following built-in stream identifiers for common use cases. All the built-in identifiers specify a sample count of 1 and an interval of 1 minute. Additionally, they sort the data on the
REQUESTS attribute. They differ only in being associated with different built-in selectors. Each built-in identifier is associated with a built-in selector of the same name (for example, the built in identifier Top_URL is associated with the built-in selector
Top_URL). Following are the built-in identifiers:

- Top_URL
- Top_CLIENTS
- Top_URL_CLIENTS_LBVSERVER
- Top_URL_CLIENTS_CSVSERVER
- Top_MSSQL_QUERY_DB_LBVSERVER
- Top_MYSQL_QUERY_DB_LBVSERVER

For more information about the built-in selectors, see Configuring a Selector.

> **Note**:
>
> The maximum length for storing string results of selectors (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, of which 50 characters are enough to uniquely identify a string, use an expression to extract only the required 50 characters.

You cannot modify a built-in identifier's configuration. However, you can create an identifier with a configuration of your choice.

To configure a stream identifier by using the command line interface

At the command prompt, type the following commands to configure a stream identifier and verify the configuration:

- `add stream identifier <name> <selectorName> [-interval <positive_integer>] [-SampleCount <positive_integer>] [-sort <sort>]`
- `show stream identifier <name>`

Example

```
> add stream identifier myidentifier Top_URL -interval 10 -sampleCount
   100
 Done
<!--NeedCopy-->
```

To configure a stream identifier by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. In the details pane, do one of the following:
    - To create a stream identifier, click **Add**.
    - To modify a stream identifier, select the identifier, and then click **Edit**.
3. In the Configure Stream Identifier page, set the following parameters:
    - Name
    - Selector
    - Interval
    - Sample Count
    - Sort
4. Click **OK**, and then click **Close**.

View statistics

September 21, 2020

You can view the collected statistics in tabular format in the command-line interface and in graphical format in the GUI.

The following table describes the collected statistics:

Statistics	**stat stream identifier** `<identifier name>` **command**	**Description**
Number of requests	Req	The number of requests for which records were created in the last `<interval>` number of minutes.

Statistics	**stat stream identifier** `<identifier name>` **command**	**Description**
Bandwidth consumed	BandW	The total bandwidth consumed by the requests that were received in the last `<interval>` number of minutes. The total bandwidth of a request is the bandwidth consumed by the request and its response. The value is rounded off to the next higher or next lower integer value. Consequently, it might differ slightly from the expected value. For example, if a request's total bandwidth consumption is 2.2 KB, one instance of the request might be shown as having consumed 2 KB and two instances might be shown as having consumed 4 KB, but three instances might be shown as having consumed 7 KB.
Response time	RspTime	The average response time for all the requests received in the last `<interval>` number of minutes.
Concurrent connections	Conn	The total number of concurrent connections that are currently open.

To view the statistical data collected for a stream identifier by using the command line

At the command prompt, type:

```
stat stream identifier <name> [<pattern> ...] [-detail] [-fullValues] [-
```

```
ntimes <positive_integer>] [-logFile <input_filename>] [-sortBy <sortBy> [<
sortOrder>]
```

Examples

Example 1 show how to sort the output on the BandW column, in the descending order. Example 2 sorts the output in Example 1, on the Req column, and in the ascending order

Example 1:

```
> stat stream identifier myidentifier -sortBy BandW Descending -fullValues
Stream Session statistics
                   Req                BandW
User1              508               125924
User2             5020                12692
User3             2025                 4316

               RspTime                 Conn
User1             5694                    0
User2              109                    0
User3                3                    0
 Done
<!--NeedCopy-->
```

Example 2:

```
> stat stream identifier myidentifier -sortBy Req Ascending  -fullValues
Stream Session statistics
                   Req                BandW
User1              508               125924
User3             2025                 4316
User2             5020                12692

               RspTime                 Conn
User1             5694                    0
User3                3                    0
User2              109                    0
 Done
<!--NeedCopy-->
```

To view the statistical data collected for a stream identifier by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.

2. Select the stream identifier whose sessions you want to view, and then click Statistics For information about how you can group the output on the basis of the values collected for various selector expressions.

Grouping records on attribute values

September 21, 2020

Statistical information such as the number of times a particular URL has been accessed overall and per client, and the total number of GET and POST requests per client can provide valuable insights into whether any of your resources need to be expanded to meet the demand or be optimized for delivery. To obtain such statistics, you must use an appropriate set of selector expressions, and then use the pattern parameter in the stat stream identifier command. The grouping is based on the pattern that is specified in the command. Grouping can be performed concurrently on the values of multiple expressions.

In the command-line interface, you can group the output by using patterns of your choice. In the GUI, the pattern depends on the choices you make when drilling down through the values of various selector expressions. For example, consider a selector that has the expressions HTTP.REQ.URL, CLIENT.IP.SRC, and HTTP.REQ.LB_VSERVER.NAME, in that order. The statistics home page displays icons for each of these expressions. If you click the icon for CLIENT.IP.SRC, the output is based on the patterns ?. The output displays statistics for each client IP address. If you click an IP address, the output is based on the patterns * `<IP address>` ? and ? `<IP address>` * where `<IP address>` is the IP address you selected. In the resulting output, if you click a URL, the pattern used is `<URL> <IP address> ?`.

To group the records on the values of selector expressions by using the command line interface

At the command prompt, enter the following command to group the records on the basis of a selector expression:

```
stat stream identifier <name> [<pattern> ...]
```

Examples

Each example uses a different pattern to demonstrate the effect of the pattern on the output of the stat stream identifier command. The selector expressions are HTTP.REQ.URL and HTTP.REQ.HEADER("UserHeader"), in that order. The requests contain a custom header whose name is UserHeader. Note that in the examples, a given statistical value changes as determined by the grouping, but the sum total of the values for a given field remains the same.

Example 1

In the following command, the pattern used is ? ?. The appliance groups the output on the values collected for both selector expressions. The row headers consist of the expression values separated by a question mark (?). The row with the header /mysite/mypage1.html?Ed displays statistics for requests made by user Ed for the URL /mysite/mypage1.html.

```
> stat stream identifier myidentifier ? ? -fullValues
Stream Session statistics
                                        Req                 BandW
/mysite/mypage2.html?Grace                1                  2553
/mysite/mypage1.html?Grace                2                     4
/mysite/mypage1.html?Ed                   8                    16
/mysite/mypage2.html?Joe                  1                  2554
/mysite/mypage1.html?Joe                  5                    10
/mysite/?Joe                              1                     4

                                    RspTime                  Conn
/mysite/mypage2.html?Grace                0                     0
/mysite/mypage1.html?Grace                0                     0
/mysite/mypage1.html?Ed                   0                     0
/mysite/mypage2.html?Joe                  0                     0
/mysite/mypage1.html?Joe                  0                     0
/mysite/?Joe                              6                     0
 Done
<!--NeedCopy-->
```

Example 2

In the following command, the pattern used is * ?. The appliance groups the output on the values accumulated for the second expression HTTP.REQ.HEADER(“UserHeader”). The rows display statistics for all requests made by users Grace, Ed, and Joe.

```
> stat stream identifier myidentifier * ?
Stream Session statistics
                 Req    BandW  RspTime      Conn
Grace              3     2557        0         0
Ed                 8       16        0         0
Joe                7     2568        6         0
 Done
<!--NeedCopy-->
```

Example 3

In the following command, the pattern used is ? *, which is the default pattern. The output is grouped on the values collected for the first selector expression. Each row displays statistics for one URL.

```
> stat stream identifier myidentifier ? * -fullValues
Stream Session statistics
                                        Req                BandW
/mysite/mypage2.html                      2                 5107
/mysite/mypage1.html                     15                   30
/mysite/                                  1                    4

                                     RspTime                Conn
/mysite/mypage2.html                       0                    0
/mysite/mypage1.html                       0                    0
/mysite/                                   6                    0
 Done
<!--NeedCopy-->
```

Example 4

In the following command, the pattern used is `* *`. The appliance displays one set of collective statistics for all the requests received, with no row title.

```
> stat stream identifier myidentifier * *
Stream Session statistics
                 Req    BandW  RspTime     Conn
                  18     5141        6        0
 Done
<!--NeedCopy-->
```

Example 5

In the following command, the pattern is `/mysite/mypage1.html *`. The appliance displays one set of collective statistics for all the requests received for the URL `/mysite/mypage1.html`, with no row title.

```
> stat stream identifier myidentifier /mysite/mypage1.html *
Stream Session statistics
                 Req    BandW  RspTime     Conn
                  15       30        0        0
 Done
<!--NeedCopy-->
```

To group the records on the values of selector expressions by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. In the details pane, click the stream identifier for which you want to view statistics, and then click **Stream Sessions**.

3. On the Home page, click the icon for the stream selector by which you want to group the output.
4. To return to the Home page from the statistics page for a selector expression, click **Home**.
5. To view statistics for the value of a given selector expression, click the value. You can repeat this step for a selector expression value in each subsequent output until you obtain the statistics you want.

Clearing a stream session

September 21, 2020

You can flush all the records that have been accumulated for a stream identifier.

To clear a stream session by using the command line interface

At the command prompt, enter the following commands to clear a stream session and verify the results:
git commit -m "test"

- clear stream session
- stat stream identifier
- one more

Example

This example uses the stat stream identifier command first, so that a comparison can be made with the stat stream identifier command that is used for verifying the result of the clear stream session command.

```
>stat stream identifier myidentifier
Stream Session statistics
                 Req     BandW  RspTime     Conn
/aed....html       2         0        0        0
/                636       303       12        0
 Done
>clear stream session myidentifier
 Done
>stat stream identifier myidentifier
 Done
<!--NeedCopy-->
```

To clear a stream session by using the GUI

1. Navigate to **AppExpert** > **Action Analytics** > **Stream Identifiers**.
2. Select the stream identifier whose sessions you want to clear, and then click **Clear Sessions**.

Configure policy for optimizing traffic

September 21, 2020

To put the selector-identifier pair in your action analytics configuration into effect, you must associate the pair with the point in the traffic flow at which you want to collect statistics. You can do so by configuring a default syntax policy and referencing the stream identifier from the policy rule. You can use compression policies, caching policies, rewrite policies, application firewall policies, responder policies, and any other policies whose action is based on a Boolean expression.

The action analytics feature introduces a set of default syntax expressions and functions for collecting and evaluating data. The expression `ANALYTICS.STREAM(<identifier_name>)` is used for referencing the identifier that you want to use. The expression COLLECT_STATS is used to collect statistical data. Functions such as `IS_TOP(<uint>)` and `IS_TOP_FREQUENTS(<uint>)` are used for making automatic, real-time traffic optimization decisions.

- `IS_TOP(<number>)`. Finds if a given object is in the top `<number>` of elements. For example, is the element among the top 10 elements. When multiple elements have the count, they are considered to be similar in nature. The sort function must be turned on to avoid an undef condition.

- `IS_TOP_FREQUENTS(<frequency>)`. Finds if a given object is in the top `<frequency>` of the elements that are in the top elements. For example, is the element among the top 50% of all the top elements maintained. Elements having the same values are considered similar in nature. The sort function must be turned on to avoid an undef condition.

It is your policy configuration that determines whether the NetScaler appliance must only collect data from traffic or also perform an action. If the appliance must only collect statistical data, you can configure a policy with the rule `ANALYTICS.STREAM(<identifier_name>).COLLECT_STATS` and the action NOOP. The NOOP policy must be the policy with the highest priority at the bind point. This policy is sufficient if you are only collecting statistics. Traffic optimization decisions, such as what to compress or cache, must be based on manual, periodic evaluation of the statistical data.

If, in addition to collecting statistics, the appliance must also perform an action on the traffic, you must configure the gotoPriorityExpression parameter of the NOOP policy such that another policy that has the desired rule and action is evaluated subsequently. This second policy must have a rule that begins with the `ANALYTICS.STREAM(<identifier_name>)` prefix and a function that evaluates the data.

Following is an example of two responder policies that are configured and bound globally. The policy `responder_stat_collection` enables the appliance to collect statistics based on the identifier, myidentifier. The policy `responder_notify` evaluates the data that is collected.

Example

```
> add responder action send_notification respondwith '"You are in the Top 10 list for bandwidth consumption"'
 Done
> add responder policy responder_stat_collection' ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
Done
> add responder policy responder_notify 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.IS_TOP(10)' send_notification
 Done
> bind responder global responder_stat_collection 10 NEXT
 Done
> bind responder global responder_notify 20 END
 Done
<!--NeedCopy-->
```

How to limit bandwidth consumption for user or client device

April 27, 2021

Your web site, application, or file hosting service has finite network and server resources available to it to serve all its users. One of the most important resources is bandwidth. Substantial bandwidth consumption by only a subset of the user base can result in network congestion and reduced resource availability to other users. To prevent network congestion, you might have to limit a client's bandwidth consumption by using temporary service denial techniques such as responding to a client request with an HTML page if it has exceeded a preconfigured bandwidth value over a fixed time period leading up to the request.

In general, you can regulate bandwidth consumption either per client device or per user. This use case demonstrates how you can limit bandwidth consumption per client to 100 MB over a time period of one hour. The use case also demonstrates how you can regulate bandwidth consumption per user to 100 MB over a time period of one hour, by using a custom header that provides the user name. In both cases, the tracking of bandwidth consumption over a moving time period of one hour is achieved by setting the interval parameter in the stream identifier to 60 minutes. The use cases also demonstrate how you can import an HTML page to send to a client that has exceeded the limit. Importing an HTML page not only simplifies the configuration of the responder action in these use cases, but also simplifies the configuration of all responder actions that need the same response.

To limit bandwidth consumption per user or client device by using the command line interface

In the command-line interface, perform the following tasks to configure action analytics for limiting a client's or user's bandwidth consumption. Each step includes sample commands and their output.

1. **Set up your load balancing configuration.** Configure load balancing virtual server mysitevip, and then configure all the services that you need. Bind the services to the virtual server. The following example creates ten services and binds the services to mysitevip.

   ```
   > add lb vserver mysitevip HTTP 192.0.2.17 80
    Done
   > add service service[1-10] 192.0.2.[240-249] HTTP 80
   service "service1" added
   service "service2" added
   service "service3" added
   .
   .
   .
   service "service10" added
    Done
   > bind lb vserver vserver1 service[1-10]
   service "service1" bound
   service "service2" bound
   service "service3" bound
   .
   .
   .
   service "service10" bound
    Done
   <!--NeedCopy-->
   ```

2. **Configure the stream selector.** Configure one of the following stream selectors:

   - To limit bandwidth consumption per client, configure a stream selector that identifies the client IP address.

     ```
     > add stream selector myselector CLIENT.IP.SRC
      Done
     <!--NeedCopy-->
     ```

   - To limit bandwidth consumption per user on the basis of the value of a request header that provides the user name, configure a stream selector that identifies the header. In the following example, the name of the header is UserHeader.

```
> add stream selector myselector HTTP.REQ.HEADER( “UserHeader” )
 Done
<!--NeedCopy-->
```

3. **Configure a stream identifier.** Configure a stream identifier that uses the stream selector. Set the interval parameter to 60 minutes.

```
> add stream identifier myidentifier myselector -interval 60 -sampleCount 1 -sort BANDWIDTH
 Done
<!--NeedCopy-->
```

4. **Configure the responder action.** Import the HTML page that you want to send to users or clients that have exceeded the bandwidth consumption limit, and then use the page in responder action crossed_limits.

```
> import responder htmlpage http://192.0.2.20:80/stdpages/wait.html crossed-limits.html
This operation may take some time, Please wait...

 Done
> add responder action crossed_limits respondwithhtmlpage crossed-limits.html
 Done
<!--NeedCopy-->
```

5. **Configure the responder policies.** Configure responder policy myrespol1 with the rule ANALYTICS.STREAM(“myidentifier”).COLLECT_STATS and the action NOOP. Then, configure policy myrespol2 for determining whether a client or user has crossed the 100 MB limit. The policy myrespol2 is configured with the responder action crossed_limits.

```
> add responder policy myrespol1 'ANALYTICS.STREAM("myidentifier").COLLECT_STATS' NOOP
 Done
> add responder policy myrespol2 'ANALYTICS.STREAM("myidentifier").BANDWIDTH.GT(104857600)' crossed_limits
 Done
<!--NeedCopy-->
```

6. **Bind the responder policies to the load balancing virtual server.** The policy myrespol1, which only collects statistical data, must have the higher priority and a GOTO expression of NEXT.

```
> bind lb vserver mysitevip -policyName myrespol1 -priority 1 -gotoPriorityExpression NEXT
 Done
> bind lb vserver mysitevip -policyName myrespol2 -priority 2 -gotoPriorityExpression END
 Done
<!--NeedCopy-->
```

7. **Test the configuration.** Test the configuration by sending test HTTP requests, from multiple clients or users, to the load balancing virtual server and using the stat stream identifier command to view the statistics that are collected for the specified identifier. The following output displays statistics for clients.

```
> stat stream identifier myidentifier -sortBy BandW  - fullValues
Stream Session statistics
                          Req           BandW
192.0.2.30                5000          3761
192.0.2.31                29            2602
192.0.2.32                25            51

                          RspTime       Conn
192.0.2.30                2             0
192.0.2.31                0             0
192.0.2.32                0             0
 Done
>
<!--NeedCopy-->
```

AppExpert applications and templates

March 10, 2021

An AppExpert application is a collection of configuration information that you set up on the NetScaler appliance for securing and optimizing traffic for a Web application, such as Microsoft SharePoint. Managing AppExpert applications is simplified by a graphical user interface (GUI) that allows you to specify application traffic subsets and a distinct set of security and optimization policies for processing each traffic subset. Additionally, it consolidates all deployment tasks in one view, so you can quickly configure target IP addresses for clients and specify host servers.

To get started with an AppExpert application, you must first obtain the appropriate application template and import the template to the NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components, statistics, and the Application Visualizer. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

AppExpert application terminology

Following are the terms used in the AppExpert applications feature and the descriptions of the entities for which the terms are used:

Public endpoint. The IP address and port combination at which the NetScaler appliance receives client requests for the associated web application. A public endpoint can be configured to receive either HTTP or secure HTTP (HTTPS) traffic. All client requests for the web application must be sent to a public endpoint. An AppExpert application can be assigned multiple endpoints. You configure public endpoints after you import a template.

Application unit. An AppExpert application entity that processes a subset of web application traffic and load balances a set of services that host the associated content. The subset of traffic that an application unit must manage is defined by a rule. Each application unit also defines its own set of traffic optimization and security policies for the requests and responses that it manages. The NetScaler services associated with these policies are Compression, Caching, Rewrite, Responder, and application firewall.

By default, every AppExpert application with at least one application unit includes a default application unit, which cannot be deleted. The default application unit is not associated with a rule for identifying requests and is always placed last in the order of application units. It defines a set of policies for processing any request that does not match the rules that are configured for the other application units, thereby ensuring that all client requests are processed.

Application units and their associated rules, policies, and actions are included in AppExpert application templates.

Service. The combination of the IP address of the server that hosts the web application instance and the port to which the application is mapped on the server, in the format <IP address>:<Port>. A web application that serves a large number of requests is usually hosted on multiple servers. Each server is said to host an instance of the web application, and each such instance of the web application is represented by a service on the NetScaler appliance. Services are deployment-specific and are therefore not included in templates. You must configure services after you import a template.

Application unit rule. Either a classic expression or a default syntax expression that defines the characteristics of a traffic subset for an application unit. The following example rule is a default syntax expression that identifies a traffic subset that consists of four image types:

HTTP.REQ.UR	HTTP.REQ.UR	HTTP.REQ.UR	HTTP.REQ.URL.SUFFIX.E

For more information about default syntax expressions and classic policy expressions, see Policies and Expressions.

Traffic Subset. A set of client requests that require a common set of traffic optimization and security policies. A traffic subset is managed by an application unit and is defined by a rule.

How AppExpert Application Works

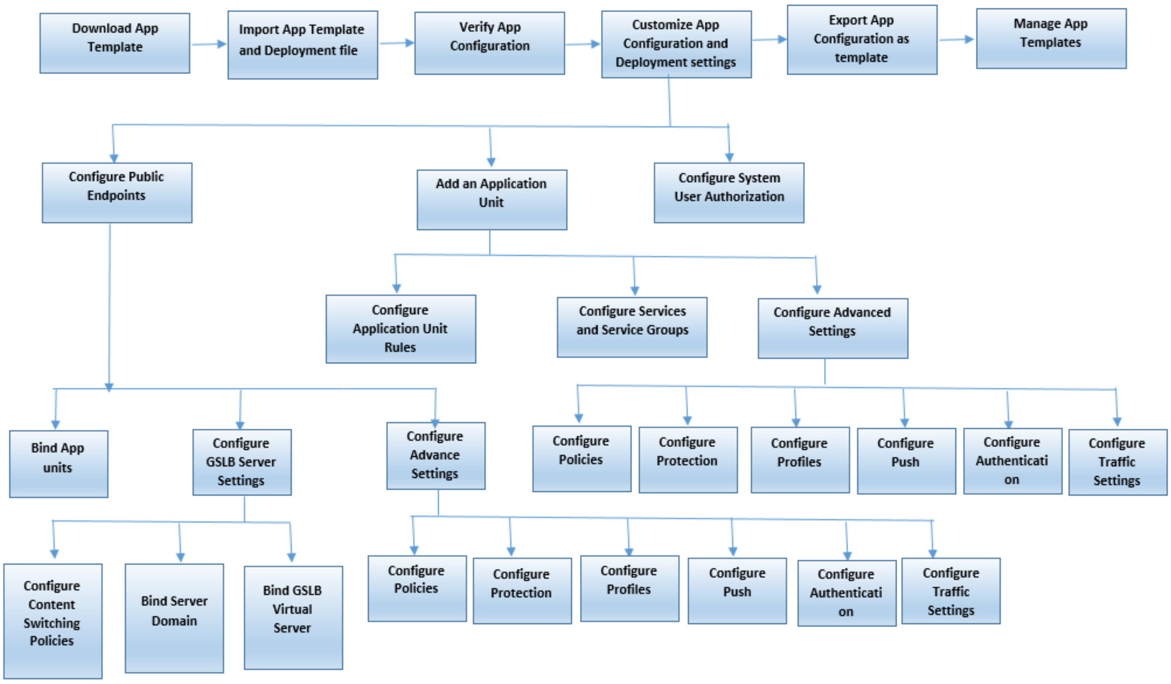
September 21, 2020

When the endpoint receives a client request, the NetScaler appliance evaluates the request against the rule that is configured for the topmost application unit. If the request satisfies this rule, the request is processed by the policies that are configured for the application unit, and then forwarded to a service. The choice of service depends on which services are configured for the application, and on settings such as the load balancing algorithm and persistence method configured for the application unit.

If the request does not satisfy the rule, the request is evaluated against the rule for the next topmost application unit. In this order, the request is evaluated against each application unit rule until the request satisfies a rule. If the request does not satisfy any of the configured rules, it is processed by the default application unit, which is always the last application unit.

You can configure multiple public endpoints for an AppExpert application. In such a configuration, by default, each application unit processes requests received by all the public endpoints and load balances all the services that are configured for the application. However, you can specify that an application unit processes traffic from only a subset of the public endpoints and load balances only a subset of the services that are configured for the AppExpert application.

The following flow diagram illustrates the AppExpert Application flow sequence for using a built-in application template.


Download App Template
Import App Template and Deployment file
Verify App Configuration
Customize App Configuration and Deployment settings
Export App Configuration as template
Manage App Templates
Configure Public Endpoints
Add an Application Unit
Configure System User Authorization
Configure Application Unit Rules
Configure Services and Service Groups
Configure Advanced Settings
Bind App units
Configure GSLB Server Settings
Configure Advance Settings
Configure Policies
Configure Protection
Configure Profiles
Configure Push
Configure Authentication
Configure Traffic Settings
Configure Content Switching Policies
Bind Server Domain
Bind GSLB Virtual Server
Configure Policies
Configure Protection
Configure Profiles
Configure Push
Configure Authentication
Configure Traffic Settings

If you prefer to create a customized application without using a template, do the following:

1. Create a custom application.
2. Configure application and deployment settings.
3. Export the configuration to new template files (optional).
4. Import the template files to other NetScaler appliances that require a similar AppExpert application configuration

Get started with AppExpert

September 21, 2020

To get started with an AppExpert application, you must first obtain an application template and import the template to a NetScaler appliance. After the AppExpert application is set up, you must verify that the application is working correctly. If required, you can customize the configuration to suit your requirements.

Periodically, you can verify and monitor the configuration by viewing the hit counters for various application components. You can also configure authentication, authorization, and auditing (AAA) policies for the application.

The process of setting up an application can be done in two ways:

- Using a prebuilt application template

- Creating a custom application without using a template.

If you prefer to set up the application by using a prebuilt application template, do the following:

1. Download an application template.
2. Import template files to NetScaler appliance.
3. Verify application setup.
4. Configure application and deployment settings.
5. Export the configuration to new template files (optional).
6. Import the template files to other NetScaler appliances that require a similar AppExpert application configuration.

NetScaler's video tutorials enable you to understand NetScaler features in easy and simple way. Watch https://www.youtube.com/watch?v=aqayflvCR_0 video to learn how to set up an application using AppExpert Application template.

Downloading an application template

April 29, 2021

AppExpert application setup begins with downloading an application template from the Citrix community website at http://community.citrix.com to your local computer or NetScaler appliance. The application templates are imported and exported, so that you can easily share application-specific configurations within an organization or across organizations. An application template includes the following set of entities:

1. Application components (for example, webpages, files, archives, and web services)
2. Traffic management entities (for example, virtual server IP addresses and associated load-balancing algorithms, and SSL offload settings) for the application components.
3. NetScaler policies used for optimizing the application traffic.

> **Note**:
>
> Application templates are available in different versions for configuring different types of NetScaler appliances.

Importing an Application Template

September 21, 2020

For NetScaler software version 9.3 or later, each AppExpert template has two XML files: a Template file and a Deployment file. You must import both files from your local computer to a NetScaler appliance.

You can either import the template files from your computer to the AppExpert application templates directory in NetScaler appliance or upload files to a NetScaler appliance and then import them from the appliance.

Note: When you import a template from an appliance, you have to provide the variable value available in the template. By default, the pre-configured value is display `https://docs.citrix.com/en-us/netscaler//11/appexpert/appexpert-application-templates/creating-managing-templates/netscaler-application-template-deployment-files.html`

After you import the template files, the application-configuration and deployment information populates the target application automatically. The appliance imports all the configuration from the template files through the NITRO API. If you do not import the deployment file, the system generates an application populated with content switching virtual server configuration. For more information about the format of application templates and deployment files, see Understanding NetScaler Application Templates and Deployment Files.

When you import a template, if you do not include a deployment file, you have to configure the public endpoints in the application that the system automatically generates from the template. One endpoint for HTTP and another endpoint for HTTPS. When configuring a public endpoint of type HTTPS, make sure you enable the SSL feature, bind the server certificate, and include the server-certificate and certificate-key files.

For more information about configuring endpoints after you import a template, see Configuring Public Endpoints.

To import AppExpert application template files to a NetScaler appliance by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click **Import Template**.
3. On the **Import** page, set the following parameters:
   a) Application Name (mandatory)
   b) Template File (mandatory)
   c) Use deployment file
4. Click **Continue** to auto populate application-configuration and deployment information into an application.

NetScaler's video tutorials enable you to understand NetScaler features in an easy and simply way. Watch https://www.youtube.com/watch?v=AR9TwSD9uJM video to learn how to import an application template.

Verifying and Testing Application Configuration

September 21, 2020

The GUI includes icons that indicate the states of the entities in the AppExpert application. These icons are displayed for applications and application units and are based on the health checks that the NetScaler appliance performs periodically on services and entities. The following table lists the icons and describes their meanings.

Icon	Entity	Indicates that
🟢	Application	At least one public endpoint is up. The application will accept client requests from the public endpoints that are up.
🟢	Application unit	The application unit is up. The application unit is up when at least one service or service group is up.
🟡	Application	The public endpoint is out of service (disabled). This indicator is displayed when only one public endpoint is configured for the AppExpert application.
🔴	Application	All the endpoints that are configured for the application are out of service. This indicator is displayed only when multiple endpoints are configured for the application.
🔴	Application unit	All the services configured for the application unit are down.

You must ensure that the icons for each application and its application units are green at all times. If the icon that is displayed for an application is not green, verify that you have configured the public endpoints correctly. If the icon that is displayed for an application unit is not green, verify that the services are configured correctly. However, note that a green indicator does not mean that the state of all associated entities is UP. It only means that the application has sufficient resources (endpoints and services) to serve client requests. To verify that the state of all associated entities is UP, check the health of all the entities on the statistics page for the application.

Customizing the configuration

September 21, 2020

After you verify that the AppExpert application is working correctly, you can customize the configuration to suit your requirements.

After you verify that the AppExpert application configuration is working correctly, you can configure the application and the deployment settings to suit your requirements. When you import an application template and deployment file, the system automatically populates the target application with the available configuration settings (such as application units, application unit rules, policies, persistence settings, load balancing methods, profiles, and traffic settings). In this application, you can configure deployment settings such as public endpoints, services, and service groups for each traffic subset. If you want the AppExpert application to manage a traffic subset that is not included in the template, you can either add an application unit for a traffic subset or modify the existing application unit. After you customize the configuration, you can also specify the order of evaluation for each traffic subset that the application manages.

Configuring an AppExpert application consists of the following steps:

1. Configuring Public Endpoints
2. Configuring Application Units
3. Specifying the Order of Evaluation
4. Viewing Application Configuration using Visualizer

Also, you can configure the policies that the template provided. If the AppExpert application template does not include policies for a particular NetScaler feature, such as Rewrite or application firewall, you can configure your own policies.

Configure public endpoints

September 21, 2020

If you did not specify a public endpoint when importing an AppExpert application, you can specify public endpoints after you create the application. You can configure one public endpoint of type HTTP and one public endpoint of type HTTPS for your AppExpert application.

If endpoints are already configured for the application, you can dissociate endpoints from the AppExpert application and delete any endpoints that you no longer need. Note that when you dissociate a public endpoint from the AppExpert application, the endpoint is automatically unbound from the associated application unit, but it is not deleted from the system.

To configure public endpoints for an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Edit.
3. In the **Applications** page, go to **Public Endpoint** section and click the pencil icon.
4. In the **Public Endpoint** slider, set the following parameters.
    a) Public endpoint type. Select the radio button to define the endpoint type.
    b) Name. Name of the public endpoint.
    c) IP address. IP address of the public endpoint.
    d) Port. Port number of the public endpoint.
    e) Protocol. Select a protocol type as HTTP or HTTPS.
5. Click **Continue.**
6. In the **Application Units** section, select an application unit from the list.
7. Click **Continue** to set the policy and server details.
8. Click **OK** and then Done.
9. Click Close.

For more information about the parameters in the **Configure Public Endpoint** dialog box, see Content Switching.

Configure services and service groups for an application unit

September 21, 2020

When you configure a service or service group, you either modify an existing service or service group, or add new services to the AppExpert application. You add services or service groups if you did not specify them when you imported the application template. You also add services and service groups when you increase the number of servers that host instances of the application. You can configure a service and service group for an application unit only after you configure the service or service group for the AppExpert application.

To configure a service or service group for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application and then click **Edit.**
3. In the **Applications** page, select an application unit and then click **Continue**.
4. In the **Services and Service Groups** section, do the following:
    a) In the Service Binding slider, set the following parameters.
        i. Service. Select a load balancing service from the list or create a new service.
        ii. Weight. Provide a weight value for the service.
    b) Click **Bind** and then **Done.**
    c) In the ServiceGroup Binding slider, set the following parameters:

        i. Service Group Name. Select a load balancing service group or create a new service group.
        ii. Click **Bind** and then **Done**.
    d) Click **Done**.
5. Click **Continue** to set other configurations.

Create application units

September 21, 2020

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications** page, go to **Application Units** section and click the **pencil** icon.

To configure policy expressions for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications **page, go to** Application Units** section and click the **+ **icon. to create a unit and add policy expressions.
4. To specify the format of the new expression, do one of the following:
    a) To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
    b) To specify that you want to configure an advanced expression in the Rule box, click Default Syntax.
    c) In the Rule box, configure the expression.
5. Click **OK**.

Configuring application unit rules

September 21, 2020

You might want to configure an application unit rule to include or exclude certain types of traffic. When you configure the rule, you can also define the syntax of the expression.

To configure an application unit rule:

1. In the navigation pane of the GUI, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to modify the rule, and then click **Open**.
3. In the **Configure Application Unit** dialog box, do the following:
    a) To specify the format of the new expression, do one of the following:
        - To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
        - To specify that you want to configure an advanced expression in the Rule box, click **Default Syntax**.
    b) In the Rule box, configure the expression.
4. Click **OK**.

Configuring policies for application units

September 21, 2020

For an AppExpert application, you can configure policies for Compression, Caching, Rewrite, Responder, and Application Firewall. The templates that you download from the Citrix Community web site provide you with a set of policies that fulfill the most common application management requirements. You might want to fine-tune or customize these policies. If the set of policies provided for a given application unit does not include policies for a particular feature, you can create and bind your own policies for that feature.

If you create an AppExpert application without using a template, you must configure all the policies that the web application needs.

The GUI uses various icons to indicate whether or not policies are configured for a feature. For an application unit, if a policy is configured for a given feature, an icon that represents the feature is displayed. For example, if a compression policy is configured for an application unit, a compression icon is displayed in the Compression column for the application unit. For features for which no policy is configured, an icon depicting a plus sign (+) is displayed.

> **Note**:
>
> When configuring policies for application units, you might need to configure policies and expressions that are either in the classic or
> default syntax. Additionally, when you configure

> default syntax policies, you might need to specify parameters such as Goto expressions and invoke policy banks.
> For information about configuring policies and expressions in both formats, see Policies and Expressions.

Configuring compression policies

You can use either classic policies or advanced policies to configure compression, but you cannot bind compression policies of both types to the same application unit.

To configure a compression policy for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Compression column.
3. In the Configure Compression Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - Click Switch to Default Syntax if you want to configure a default syntax compression policy. If you want to bind or configure classic compression policies, and if you are in the default syntax view, you can click Switch to Classic Syntax to return to the classic policy view and begin modifying bound classic policies or create and bind new classic compression policies.
      Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you are in the
      default syntax view, when you click
      Insert Policy, the list that appears in the
      Policy Name column will include only
      default syntax policies. You cannot bind policies of both types to an application unit.
    - If you want to configure classic policies, click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
      You can configure both request-time and response-time classic compression policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
    - To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Compression Policy dialog box, modify the policy, and then click OK.
      For information about modifying a compression policy, see Compression.
    - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
    - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
    - To regenerate assigned priorities, click **Regenerate Priorities**.

- To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Compression Policy** dialog box, configure the policy, and then click **Create**.

  For information about modifying a compression policy, see Compression.
- If you are configuring a default syntax expression, do the following:
  - In the **Goto Expression** column, select a Goto expression.
  - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click **Apply Changes**, and then click **Close**.

Configuring Caching Policies

You can use only default syntax policies and expressions to configure Caching policies.

To configure Caching policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Caching column.
3. In the **Configure Cache Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.

     You can configure both request-time and response-time Caching policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
   - To modify a Caching policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy.** Then, in the **Configure Cache Policy** dialog box, modify the policy, and then click **OK**.

     For information about modifying a Caching policy, see Integrated Caching.
   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.
   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the Policy Name column, click **New Policy**. Then, in the **Create Cache Policy** dialog box, configure the policy, and then click **Create**.

     For information about modifying a Caching policy, see Integrated Caching.
   - In the Goto Expression column, select a Goto expression.
   - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.

4. Click **Apply Changes**, and then click **Close**.

Configuring Rewrite policies

You can use only default syntax policies and expressions to configure Rewrite policies.

To configure Rewrite policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Rewrite column.
3. In the **Configure Rewrite Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
     You can configure both request-time and response-time Rewrite policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
   - To modify a Rewrite policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Rewrite Policy dialog box, modify the policy, and then click OK.
     For information about modifying a Rewrite policy, see Rewrite.
   - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
   - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
   - To regenerate assigned priorities, click **Regenerate Priorities**.
   - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Rewrite Policy** dialog box, configure the policy, and then click **Create**.
     For information about modifying a Rewrite policy, see Rewrite.
   - In the Goto Expression column, select a Goto expression.
   - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Responder policies

You can use only default syntax policies and expressions to configure Responder policies.

To configure Responder policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.

2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Responder column.
3. In the **Configure Responder Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Responder Policy** dialog box, modify the policy, and then click **OK**.
      For information about modifying a Responder policy, see Responder.
    - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
    - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
    - To regenerate assigned priorities, click **Regenerate Priorities.**
    - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name column**, click **New Policy**. Then, in the **Create Responder Policy** dialog box, configure the policy, and then click **Create**.
      For information about modifying a Responder policy, see Responder.
    - In the Goto Expression column, select a Goto expression.
    - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Application Firewall policies

You can configure both classic and default syntax policies and expressions for Application Firewall. However, if a policy of one type is already bound globally or to a virtual server that is configured on the appliance, you cannot bind a policy of the other type to an application unit. For example, if a default syntax policy is already bound either globally or to a virtual server, you cannot bind a classic policy to an application unit.

To configure Application Firewall policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Application Firewall column.
3. In the **Configure Application Firewall Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - Click either Classic Expression or Advanced Expression depending on the type of expression you want to configure for the Application Firewall policy.
      Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you select
      Advanced Expression, when you click

  Insert Policy, the list that appears in the
  Policy Name column will include only
  default syntax policies. You cannot bind policies of both types to an application unit. This option is not available if a policy of either type is already bound either globally or to a virtual server.
- To modify an application firewall policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Application Firewall Policy dialog box, modify the policy, and then click OK.
  For information about modifying a application firewall policy, see Policies.
- To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
- To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
- To regenerate assigned priorities, click **Regenerate Priorities**.
- To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Application Firewall Policy** dialog box, configure the policy, and then click **Create**.
  For information about modifying a application firewall policy, see Policies.

4. Click **Apply Changes**, and then click **Close**.

Configuring Application Units

September 21, 2020

To configure an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications** > **Application Unit** section and then click the plus icon to add a new application unit for a traffic subset.
2. In the **Application Unit** slider, set the following parameters:
   - Name
   - Expression

   You can insert an expression either by adding the expression components manually or by using the Expression Editor link. To manually add an expression, enter a selector component and then type a period (.) to display a list from which you can select the next component. For example, type HTTP and then type a period. A drop-down menu appears. The contents of this menu provide the keywords that can follow the initial keyword that you entered. Select a component from the drop-down menu. The Expression* text box now displays the components that you have added to the expression (for example, HTTP.REQ). Continue adding components until the complete expression is formed.

If you prefer assistance to form the expression, you can use the Expression Editor link. On the Expression Editor page, you can form an expression by selecting components from the drop-down boxes. Select the components and click Done to insert the expression on the Application Unit page.

3. Click **Continue** to bind services and service groups.
4. Click the **Service** section to select or add a virtual service and bind it to the application unit.
5. Click **Continue** and click the **Service Group** section to select or add a virtual service group and bind it to the application unit.
6. Click **Bind** and **Continue** to configure Advanced Settings (such as Policies, Method, Persistence, Protection, Profiles, Push, Authentication, and Traffic Settings) for the application unit.
7. Click the **plus** icon in each section to set the configuration parameters.
8. Click **OK** and then **Done**.

To edit an application unit for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, select an entity, click the edit icon and modify the application unit settings.

Note: You cannot modify the name and rule expression for an existing application unit.

NetScaler's video tutorials enable you to understand NetScaler features in an easy and simple way. Watch <`https://www.youtube.com/watch?v=bJ5_i8fV2hc`> video to learn how configure an application unit.

Configuring public endpoints for an application

September 21, 2020

To configure public endpoints for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Public Endpoint** section, click **+** to configure a new public endpoint.
3. In the **Public Endpoint** slider, do one of the following:
    a) Click **New** to create a new endpoint.
    b) Click **Existing Public Endpoint** to select an endpoint from the drop-down list.
4. Set the following endpoint parameters:
    a) Name
    b) IP address
    c) Protocol
    d) Port

5. Click **Continue** to configure additional settings such as application units, GSLB server bindings, policies, profiles, push, traffic settings, and authentication.
6. Click **OK** and then **Done**.
7. Click **Continue** and then **Done**.

To edit a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application, and click **Edit**. In the **Public Endpoint** section, select an endpoint, click the pen icon, and modify the endpoint settings.

To delete a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications** > **Public Endpoint,** click the pen icon to view the delete icon next to the entity.

NetScaler's video tutorials enable you to understand NetScaler features in an easy and simply way. Watch `<https://www.youtube.com/watch?v=z4v-edQiVpw>` video to learn how to configure a public endpoint.

Specifying the Order of Evaluation of Application Units

September 21, 2020

Application unit rules are evaluated in the order in which they are placed in the GUI. The rule that is configured for the topmost application unit is always configured first, followed by the rule that is configured for the second topmost application unit, and so on. The default application unit is always evaluated last.

When a request matches the rule that is configured for an application unit, the request is processed by the application unit, and no further matching is performed. Therefore, the order of evaluation of application units becomes an important factor if the traffic subsets for two or more application units overlap. If the traffic subsets for two or more application units overlap, you must specify the order in which an incoming request is matched against the application unit rules.

To specify the order of evaluation of application units:

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the **Pencil** icon and then hover the cursor over the check box to the left of the name of the application unit. Click the icon that appears next to the check box and hold down the mouse to drag the application up or down to a new location in the priority list.

Configuring Persistency Groups for Application Units

September 21, 2020

You can configure a persistency group for the application units in an AppExpert application. In the context of an AppExpert application, a persistency group is a group of application units that you can treat as a single entity for the purpose of applying common persistence settings. When the application is exported to an application template file, the persistency group settings are included, and they are automatically applied to the application units when you import the AppExpert application.

To configure a persistency group for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the **Applications View** dialog box, click the name of the application for whose application units you want to configure a persistency group, and then click **Configure Persistency Groups**.
3. In the **Configure Persistency Groups** dialog box, do one of the following:
   - To add a persistency group, click **Add**.
   - To modify a persistency group, click **Open**.
4. In the **Create Persistency Group** or **Configure Persistency Group** dialog box, set the following parameters:
   - Group Name—Name of the persistency group. For the NetScaler appliance to recognize the persistency group as part of the application's configuration, the name of the AppExpert application must be included in the name of the persistency group, as a prefix. Therefore, by default, the appliance displays the prefix in the Group Name box, and you cannot remove that prefix. Enter a name of your choice after the prefix.
   - Persistence—Type of persistence for the virtual server. If you select SOURCEIP, in the IPv4 Netmask box, enter a network mask that specifies the number of bits that the appliance must consider when creating persistence sessions. If you select COOKIEINSERT, in the Cookie Domain and Cookie Name boxes, specify a domain attribute to send in the Set-Cookie directive, and a name for the cookie, respectively.
   - Timeout—Time period for which a persistence session is in effect.
   - Backup Persistence—Type of backup persistence for the group.
   - Backup Timeout—Time period, in minutes, for which backup persistence is in effect.
   - Application Units—To add an application unit to the persistency group, in the Available Application Units box, click the application unit, and then click **Add**. To remove an application unit from the persistency group, in the Configured Application Units box, click the application unit, and then click **Remove**.
5. Click **OK**.

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer

September 21, 2020

The Visualizer feature shows you a graphical representation of an application's configuration. It includes the name of the public endpoint, application units assigned to the public endpoint, and the number of policies and services bound to the application. You can use the Visualizer to obtain a visual overview of an AppExpert application's configuration and configure some of the displayed entities. By default, the Visualizer displays application units, services, and monitors for the selected application.

To view an AppExpert application by using the Application Visualizer:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and click **Visualizer**.

Configuring User Authentication, Authorization, and Auditing

September 21, 2020

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure AAA users and AAA user groups for an application by using the configuring utility

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Advanced Settings** section, click **Authorization**, and configure authorized users and user groups.
3. Click the **AAA** user section to bind authorized users to the application.
4. In the **AAA User** slider, set the parameters .
5. Click **Continue**, and then click **Authorization Policies** in the **Advanced Settings** section.
6. In the **Authorization Policy** slider, bind an authorization policy to the application.
7. Click **Continue**, and then click the **Authorization Group** section in the **Advanced Settings** section.
8. In the **AAA Group Binding** slider, bind an authorization user group to the application.
9. Click **Continue**, and then click **Policies** in the **Advanced Settings** section.
10. In the **Policies** slider, bind an **Audit Syslog** or **Audit NSlog** policy to the application.
11. Click **Continue** and then **Done**.

To edit AAA users and AAA user groups for an application by using the GUI:

Navigate to **AppExpert** > **Applications** > **Advanced Settings** and click **Authorization**. Then click the edit icon and specify values for user or user-group authorization settings.

To delete AAA users and AAA user groups by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Applications** page, click **Advanced Settings** and click **Authorization**. Click the delete icon next to the entity.

Monitoring a NetScaler Application

September 21, 2020

After you customize the AppExpert application, you can view application statistics to make sure that the application and all its entities are working correctly. You can also use the Application Visualizer to monitor statistics associated with certain entities such as policies and virtual servers.

You can also view the hit counters for various entities at regular intervals to make sure that counters are being updated.

Viewing Application Statistics

In the **Applications** node, you can select an application and view the Statistics page for the application. On the Statistics page, you can monitor the health and states of public endpoints and application units, and view the following statistical information:

- Requests and responses per second for each of the public endpoints and application units.
- Bytes per second, at each endpoint, for incoming and outgoing traffic.
- Application unit hit counters and the number of client and server connections for each application unit.
- Statistics for the services that are bound to the application units.

On the Statistics page, you can also view CPU usage, memory usage, and system logs.

To view statistics for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to view statistics, and then click **Statistics**.

Monitoring an Application by Using the Application Visualizer

You can use the Application Visualizer to monitor the number of requests received per second at a given point in time by the vservers and the number of hits per second at a given point in time for Rewrite, Responder, and Cache policies.

To view statistical information for vservers, Rewrite policies, Responder policies, and Cache policies in the Visualizer:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, select the application for which you want to view statistical information, and then click **Visualizer**.
3. In the **Application Visualizer** window, do the following:
    - To view the statistics, click **Show Stats**.
      The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time and has to be refreshed manually.
    - To refresh the statistical information, click **Refresh Stats**.

Viewing Hits

The hit counters that are provided for various AppExpert application entities enable you to monitor the functioning of public endpoints and application units. For an application, the Hits dialog box displays the total number of requests received by each configured public endpoint. For an application unit, the Hits dialog box displays the number of requests that the application unit processed from each of the public endpoints and the total hit count. For instructions on viewing hit counters, see Verifying and Testing the Configuration.

Deleting an Application

September 21, 2020

If you no longer need an application and its application units, you can delete it. When you delete an AppExpert application, backend services are not deleted, and any public endpoints that the application used become available for use by other applications.

When deleting an application, you are also prompted to specify whether you want to delete any bound policies and actions that are not used elsewhere.

To delete an application unit for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the delete icon next to the entity

Configure application authentication, authorization, and auditing

September 21, 2020

You can configure Authentication, Authorization, and Auditing (AAA) for the applications that you configure on the appliance. An authentication policy that is configured for an application defines the type of authentication to apply when a user or group attempts to access the application. If external authentication is used, the policy also specifies the external authentication server. Authorization policies configured for an application specify whether a particular user or group can access the application. Auditing policies define the audit log type, the level at which logging is performed, and other audit server settings. Authentication and auditing policies use the classic policy format.

Authentication policies, authorization policies, and auditing policies can be configured in any order. However, before you configure AAA for an application, you must configure a public endpoint for the application.

Configuring authentication for an application involves specifying an authentication FQDN, an authentication virtual server, a server certificate, and authentication and session policies. Authentication policies are automatically bound to the authentication virtual server specified for the application.

To configure authentication for an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, do one of the following:
    a) Click Add to add an authentication for a new application.
    b) Click Edit to modify an existing application.
3. In the **Applications** page, select an Application Unit.
4. In the **Application Unit** slider page, click Authentication from the **Advanced Settings** section.
5. In the **Authentication** section, select the authentication type as follows:
    a) Form based authentication
    b) 401 based authentication
    c) None
6. Click **OK** and then click **Done**.

Configure application authorization

You can configure authorization for users and groups to enable then to access an AppExpert application. If the AAA user or group for which you want to configure permissions has not already been created, you can create it from AppExpert and then configure permissions for application access.

To configure permissions for a AAA user or group to access an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the AppExpert application for which you want to configure a user or group access.
3. In the **Applications** page, and then click Authorization. from the **Advanced Settings** section.
4. Do one of the following:

  - If the AAA user or group for which you want to configure permissions are already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the application tree. Then, right-click the user or group and click Allow.
  - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the application tree, right-click Users or Groups, and then click Add. In the Create AAA Group or Create AAA User dialog box, fill in the values, click Create, and then click Close.
    The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
5. Click **Done** and then click **Close**.

Configure application auditing

When you configure auditing policies for an application, you must specify the server to which the log messages must be directed, the format of the messages logged, and the log level. Optionally, you can configure other settings, such as the log facility and date format. Auditing policies are automatically bound to all the AppExpert application's public endpoints.

To configure auditing policies for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to configure auditing policies.
3. In the Application Unit slider page, click + icon in the **Policies** section to configure the auditing policies.
4. In the **Policies** slider page, select policy type as Syslog auditing or Nslog auditing and click **Continue**.
5. In the Policy binding section, set the following parameters.
   a) Select a policy for binding. If you do not have a policy for binding. click + to create a new policy.
   b) To create a new auditing policy, under Policy Name, click **New Policy,** and then, in the** Polic**y page do the following:
      i. In the Name box, type a name for the policy.
      ii. The Name box already contains the string that is required at the beginning of the server name. You cannot modify the string.
      iii. From the Auditing Type list, select the auditing type (either SYSLOG or NSLOG).
      iv. If the audit server you want to specify is already listed in the Server list, select the server from the list, and then, if you want to modify the server settings, click Modify. In the Configure Auditing Server dialog box, modify the settings as appropriate, and then click OK. For more information about the settings in the Configure Auditing Server dialog box, see Auditing Authenticated Sessions.
      v. If you want to configure a new audit server, click New, and then, in the Create Audit-

        ing Server dialog box, type a name for the server, specify the server IP address, port number, and other settings as appropriate. When finished, click **OK**.
        vi. Click **Create**.
    c) To change the priorities for the new auditing policies you created, under Priority, for each policy for which you want to change the priority, double-click the priority value and type new priority value.
    d) To regenerate priorities, click **Regenerate Priorities**.
    e) To unbind a policy, click the policy, and then click **Unbind Policy**.
    f) To modify a policy, click the policy, and then click **Modify Policy.**
6. Click **Apply Changes,** and then click **Close**.

Disabling AAA for an Application

After you configure AAA for an application, you can disable the AAA configuration for that application. When you disable AAA for an application, the configuration is not lost. You can enable AAA for the application when you want to reapply the configuration.

To enable or disable AAA for an application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, click the application for which you want to enable or disable AAA, and then do one of the following:
3. To disable AAA for the application, click **Turn Off AAA.**
4. To enable AAA for the application, click **Turn On AAA**.

Setting Up a Custom NetScaler Application

September 21, 2020

If an AppExpert application template is not available for the Web application that you want to manage through the NetScaler appliance, or if available AppExpert application templates do not suit your requirements, you can create an AppExpert application without a template.

To create an AppExpert application without a template, you must first create an application and application units. Then, you configure public endpoints, services, and service groups. Finally, you configure the policies that determine how application traffic is evaluated and processed.

After you create the application and application units and configure policies, you must verify the configuration and test it to make sure that it is working correctly, just as you would when you configure an application by using a prebuilt AppExpert application template. Then, you must monitor the application to make sure that the application and its entities are working correctly.

Creating an Application

When you create an AppExpert application, the appliance creates a container to which you can add application units. The default application unit is not created until you create the first application unit.

To create an AppExpert application by using the GUI:

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click Applications, and then click Add.
3. In the Create Application dialog box, in Name, enter a name for the application, and then click OK.

Creating Application Units

For each subset of traffic associated with your web application, you must create an application unit.

To create an application unit for the AppExpert application by using the GUI:

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to add an application unit, and then click Add.
3. Click Create.

Configuring Public Endpoints for an AppExpert Application

After you have created all the application units that you require, you must configure one or more public endpoints to enable clients to access the web application through the NetScaler appliance.

To configure public endpoints for an AppExpert application by using the GUI:

1. Navigate to AppExpert > Applications.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application, do one of the following:
   - If the endpoints you want are listed in the dialog box, click the corresponding check boxes.
   - If you want to specify all the public endpoints, click Activate All.
   - If you want to dissociate endpoints from the AppExpert application, clear the corresponding check boxes.
   - If you want to create a new public endpoint, click Add. Then, in the Create public endpoint dialog box, configure endpoint settings, and then click OK.
     In the Create public endpoint dialog box, you can specify only the name, IP address, port, and protocol for the endpoint. You can specify additional endpoint settings after you create the public endpoint. To specify additional endpoint settings, after you create the endpoint, in the Choose Public Endpoints dialog box, click the endpoint, and then click Open.

  Then, in the Configure Public Endpoint dialog box, provide additional settings, and then click OK.
  For more information about the parameters in the Create public endpoint and Configure Public Endpoint dialog boxes, see Content Switching.
  - If you want to modify a public endpoint, click the endpoint, and then click Open. Then, in the Configure Public Endpoint dialog box, modify settings for the endpoint, and then click OK.
    For more information about the parameters in the Configure Public Endpoint dialog box, see Content Switching.
4. Click Close.

Configuring Public Endpoints for an Application Unit

For an application unit, you specify public endpoints in the same way as you would specify public endpoints for an application that is created from an AppExpert application template. For more information about specifying a subset of the endpoints for an application unit, see Configuring Endpoints for an Application Unit.

To configure endpoints for an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click **Configure Public Endpoints**.
3. In the **Choose Public Endpoints** dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click **OK**.

Configuring Services and Service Groups for an AppExpert Application

Services and service groups are available for application units only after you configure the services and service groups for the AppExpert application. Therefore, you must configure services and service groups for the AppExpert application before you configure the services for the application units. All the services and service groups that you configure for an AppExpert application must use the same protocol (either HTTP or HTTPS). The procedure for configuring services and service groups for an AppExpert application that is not created from a template is the same as that for an application created from a template.

To configure a service or service group for the AppExpert application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure services or service groups, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
    - To configure services, click the **Services** tab.
    - To configure service groups, click the **Service Groups** tab.
4. On the Service or **Service Groups** tab, do one of the following:
    - If the services or service groups that you want are listed on the tab, click the corresponding check boxes.
    - If you want to specify all the services or service groups, click **Activate All**.
    - If you want to create a new service or service group, click **Add**. Then, in the **Create Service** dialog box or **Create Service Group** dialog box, configure settings for the service or service group, respectively, and then click **Create**.
    - If you want to modify a service, click the service, and then click **Open**. Then, in the Configure Service dialog box or Create Service Group dialog box, configure settings for the service or service group, respectively, and then click **OK**.

For information about the settings in the Create Service, Configure Service, and Create Service Group dialog boxes, see Load Balancing.

Configuring services and service groups for an application Unit

After you configure services and service groups, you must configure services and service groups for each application unit. However, this step is not necessary if each backend service hosts all the content associated with the web application. You configure services and service groups for an application unit if the content associated with the application unit is hosted on only a subset of the backend servers.

To configure services or service groups for an application unit by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
    - To configure services, click the **Services** tab.
    - To configure service groups, click the **Service Groups** tab.
4. In the **Services** or **Service Groups** tab, do one of the following:
    - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the Weight column, specify the weight that you want to assign to each configured service.
    - To specify all services or service groups, click **Activate All**.

5. On the Method and **Persistence** and **Advanced** tabs, specify the desired parameters.
6. Click **OK**.

Configuring Policies

The procedures for configuring policies for an AppExpert application that is created without using a template are the same as those for an AppExpert application that was created from a template. For more information, see Configuring Policies for Application Units.

Creating and managing template files

September 21, 2020

After you set up an AppExpert application and customize it to suit your requirements, you can create a template from the configuration and then share the template with other administrators. Or, you can create a template and then import the template to other NetScaler appliances that require a similar AppExpert application configuration. This simplifies and expedites the process of setting up similar applications on other appliances.

AppExpert application template files can be exported either to the template directory on the NetScaler appliance or to a folder on your local computer. You can then upload and download the templates to and from the NetScaler appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

AppExpert application template files can be exported either to the template directory on the NetScaler appliance or to a folder on your local computer. You can then upload and download the templates to and from the NetScaler appliance and rename the templates that are stored in the AppExpert application templates directory on your appliance.

Exporting an appExpert application to a template file

September 21, 2020

When you export an appExpert application, all application-configuration information is exported to a template file, and all deployment-specific information is exported to a deployment file. The string _deployment is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory. If you have configured a NetScaler Gateway application, you can choose to include the NetScaler Gateway policies in the template.

To export an AppExpert application to a template file by using the GUI:

1. Navigate to **AppExpert** > **Application**, select an application entity, and then click **Edit**.
2. On the **Applications** page, click the **Export** as a Template link to export the application configuration and deployment settings as a template.
3. In the **Export Application** slider, set the following parameters:
   a) Template Filename
   b) Deployment Filename
4. Click **Continue** and **Done**.
5. Navigate to **AppExpert** > **Application** and click **Manage Templates** to show the exported configuration as files on the **Template File** and **Deployment File** tabs.

Exporting a Content Switching Virtual Server Configuration to a Template File

September 21, 2020

You can also export a content switching configuration as an application template. You can export a content switching virtual server configuration to an application template either from the Content Switching Virtual Servers pane or from the Content Switching Visualizer. Configuration information, which includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies, is exported to a template file and all deployment-specific information is exported to a deployment file. The string "_deployment" is automatically appended to the name of the template file to create the name of the deployment file. Both files are in XML format. If you choose to export the application template file to the NetScaler appliance, the template file is stored in the /nsconfig/nstemplates/applications directory on the NetScaler appliance and the deployment file is stored in the /nsconfig/nstemplates/applications/deployment_files/ directory.

For more information about the format of application templates and deployment files, see Understanding NetScaler Application Templates and Deployment Files. The configuration information that is exported includes the content switching virtual server, all associated load balancing virtual servers, services, service groups, and policies.

However, if the content switching virtual server is already configured as the public endpoint for an AppExpert application, you cannot export the configuration to a template file . In this scenario, you must export the associated AppExpert application to a template.

For more information about exporting an AppExpert application to a template file, see Exporting an AppExpert Application to a Template File.

To export a content switching configuration to an application template file from the Content Switching Visualizer by using the GUI:

1. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**.
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click **Visualizer**.
3. In the **Content Switching Visualizer**, click the icon for the content switching vserver, click **Related Tasks**, and then click **Create Template**.
4. In the **Export…as Template** dialog box, enter a name for the template file, and then do one of the following:
    - To export the template file to the appliance, make sure that Browse (Appliance) is displayed.
    - To export the template file to your computer, click the Browse (Appliance) drop-down menu, click **Local**, browse to the location to which you want to save the file, and then click **Save**.
5. Provide the following information:
    - **Introduction Description**—Any text that introduces the AppExpert application template during import. This text is displayed on the Specify Application Name page of the AppExpert Template Wizard when the template is imported.
    - **Summary Description**—Any summary that you might want to display on the Summary page of the AppExpert Template Wizard when the template is imported.
    - **Author**—The name of the author of the template.
    - **Major**—The major version number of the template.
    - **Minor**—The minor version number of the template. This number is appended to the major version number and displayed on the Summary page of the AppExpert Template Wizard, during import, in the format Major.Minor.
6. Click OK.

To export a content switching configuration to an application template file from the Content Switching Virtual Servers pane by using the GUI:

1. Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers.**
2. In the details pane, click the name of the content switching virtual server whose configuration you want to export as a template file, and then click **Create AppExpert** Template.
3. Perform steps 4 through 6 described in **To export a content switching configuration to an application template file from the Content Switching Visualizer** procedure.

Creating Variables in Application Templates

September 21, 2020

Application templates support the declaration of variables in the policy expressions and actions that are configured for an application. The ability to declare variables in policy expressions and actions enables you to replace preconfigured values in expressions (for example, configurable parameters such as the host name of a server or the target for a Rewrite action) with values that suit the environment into which you are importing the template. If variables have been configured for an AppExpert application template, the AppExpert Template Wizard, which appears when you import an AppExpert application template, includes a Specify Variable Values page on which you can specify appropriate values for the variables that are configured for the template.

As an example, consider the following policy expression that is configured to evaluate the value of the Host header in an HTTP request:

```
HTTP.REQ.HEADER("Host").CONTAINS("server1")
<!--NeedCopy-->
```

If you want the server name to be configurable at import time, you can specify the string "server1" as a variable. When importing the template, you can specify a new value for the variable on the Variables tab.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

In the export application template wizard, you can define variables in certain fields (fields with an adjacent button) for the following entities:

- Cache policies
- Rewrite policies
- Rewrite actions
- Responder policies
- Responder actions

To configure a variable in a policy expression or action by using the GUI:

1. Navigate to **AppExpert > Variables**.

2. In the **Variables** page, click **Add**.
3. In the **Create Variables** page, set the following parameters.
   Name. Name of the variable.

   Scope. Select the scope as Global or transaction.

   Type. Select the variable type as text, ulong, map.

   Expires in. Enter the expiry date.

   If Full*. Action to perform if an assignment to a map exceeds its configured max-entries:
   lru - (default) reuse the least recently used entry in the map.
   undef - force the assignment to return an undefined (Undef) result to the policy executing the assignment.
   Possible values: undef, lru
   Default value: lru.

   if no value. Value expiration in seconds. If the value is not referenced within the expiration period it will be deleted. 0 (the default) means no expiration. Minimum value: 0, Maximum value: 31622400

   Init Value. Initialization value for this variable, to which a singleton variable or map entry will be set if it is referenced before an assignment action has assigned it a value. If the singleton variable or map entry already has been assigned a value, setting this parameter will have no effect on that variable value. Default: 0 for ulong, NULL for text Maximum Length: 127

   Comments. A brief description about the variable.
4. Click **Close**.

Uploading and downloading template files

September 21, 2020

Template files can be uploaded from your local computer to the NetScaler appliance or downloaded from the appliance to your local computer. On the appliance, AppExpert application templates are always stored in the AppExpert application templates directory, which is `/nsconfig/nstemplates/applications/`.

To upload an AppExpert application template from your local computer to the NetScaler appliance:

1. Navigate to **AppExpert** > **Templates**.
2. In the details pane, click **Manage Templates**.

3. In the **Manage Application Templates** dialog box, click **Application Templates**, and then click **Upload**.
4. In the **Upload Application Template** dialog box, browse to the directory in which the template file is stored, click the template file, and then click **Select**.

   The template file is uploaded to the AppExpert application template directory on the appliance.

To download an AppExpert application template from the NetScaler appliance to your local computer:

1. Navigate to **AppExpert** > **Templates**.
2. In the details pane, click **Manage Templates**.
3. In the **Manage Application Templates** dialog box, click the AppExpert application template that you want to download, and click **Download**.
4. In the **Download Application Template** dialog box, browse to the location to which you want to save the file, and then click **Save**.

Understanding NetScaler application templates and deployment files

September 21, 2020

When you export a NetScaler application, the following two files are automatically created:

- **NetScaler application template file**. Contains application-configuration information such as application units, rules, and configured policies.
- **Deployment file**. Contains deployment-specific information such as public endpoints, services, associated IP addresses, and configured variables.

In a template file or deployment file, each unit of application-configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, each public endpoint and associated endpoint details are encapsulated within the <`appendpoint`> and <`appendpoint`> tags, and all the endpoint elements are encapsulated within the <`appendpoint_list`> and <`appendpoint_list`> tags.

Note: After you export a NetScaler application, you can add elements, remove elements, and modify existing elements before importing the application to a NetScaler appliance.

Example of a NetScaler Application Template

Following is an example of a template file that was created from a NetScaler application called **SharePoint_Team_Site**:

```
<?xml version="1.0" encoding="UTF-8" ?>
<template>
```

```
<template_info>
   <application_name>SharePoint_Team_Site</application_name>
   <templateversion_major>1</templateversion_major>
   <templateversion_minor>1</templateversion_minor>
   <author>Ed</author>
   <introduction>An application for managing a SharePoint team site
      with images, reports, and, XML content.</introduction>
   <summary>This template includes variables</summary>
   <version_major>9</version_major>
   <version_minor>3</version_minor>
   <build_number>38</build_number>
</template_info>
<apptemplate>
   <rewrite>
      <rewriteaction_list>
         <rewriteaction>
            <name>Rw_name</name>
            <type>replace</type>
            <target>HTTP.REQ.BODY(10000).AFTER_REGEX(re/number/).
               BEFORE_REGEX(re/address/)</target>
            <stringbuilderexpr>"NA"</stringbuilderexpr>
            <allow_unsafe_pi1>NO</allow_unsafe_pi1>
         </rewriteaction>
         <rewriteaction>
            .
            .
            .
         </rewriteaction>
         .
         .
         .
      </rewriteaction_list>
      <rewritepolicy_list>
         <rewritepolicy>
            <name>Rw_number_NA</name>
            <rule>HTTP.REQ.BODY(100000).CONTAINS("admin")</rule>
            <action>Rw_name</action>
         </rewritepolicy>
         <rewritepolicy>
            .
            .
            .
         </rewritepolicy>
         .
         .
```

```
            .
        </rewritepolicy_list>
    </rewrite>
    <appunit_list>
        <appunit>
            <name>SharePoint_Team_Sitedefault</name>
            <rule />
            <expressiontype>PE</expressiontype>
            <servicetype>HTTP</servicetype>
            <ipv46>0.0.0.0</ipv46>
            <ipmask>*</ipmask>
            <port>0</port>
            <range>1</range>
            <persistencetype>NONE</persistencetype>
            <timeout>2</timeout>
            <persistencebackup>NONE</persistencebackup>
            <backuppersistencetimeout>2</backuppersistencetimeout>
            <lbmethod>LEASTCONNECTION</lbmethod>
            <persistmask>255.255.255.255</persistmask>
            <v6persistmasklen>128</v6persistmasklen>
            <pq>OFF</pq>
            <sc>OFF</sc>
            <m>IP</m>
            <datalength>0</datalength>
            <dataoffset>0</dataoffset>
            <sessionless>DISABLED</sessionless>
            <state>ENABLED</state>
            <connfailover>DISABLED</connfailover>
            <clttimeout>180</clttimeout>
            <somethod>NONE</somethod>
            <sopersistence>DISABLED</sopersistence>
            <redirectportrewrite>DISABLED</redirectportrewrite>
            <downstateflush>DISABLED</downstateflush>
            <gt2gb>DISABLED</gt2gb>
            <ipmapping>0.0.0.0</ipmapping>
            <disableprimaryondown>DISABLED</disableprimaryondown>
            <insertvserveripport>OFF</insertvserveripport>
            <authentication>OFF</authentication>
            <authn401>OFF</authn401>
            <push>DISABLED</push>
            <pushlabel>none</pushlabel>
            <l2conn>OFF</l2conn>
        </appunit>
        <appunit>
            .
            .
            .
        </appunit>
        .
        .
        .
    </appunit_list>
    </apptemplate>
    <parameters>
        <property_list>
            <property>
                <variable_definition_list>
                    <variable_definition>
                        <name>body_size</name>
                        <defaultvalue>10000</defaultvalue>
                        <description>Evaluation Scope</description>
                        <startindex>14</startindex>
                        <length>5</length>
                    </variable_definition>
                    .
                    .
                    .
                </variable_definition_list>
                <object_type>rewriteaction</object_type>
                <object_name>Rw_name</object_name>
                <name>target</name>
            </property>
            .
            .
            .
        </property_list>
    </parameters>
</template>
<!--NeedCopy-->
```

Example of a deployment file

Following is the deployment file associated with the **SharePoint_Team_Site** application in the preceding example:

```
<?xml version="1.0" encoding="UTF8" ?>
<template_deployment>
    <template_info>
        <application_name>SharePoint_Team_Site</application_name>
```

```
      <templateversion_major>1</templateversion_major>
      <templateversion_minor>1</templateversion_minor>
      <author>Ed</author>
      <introduction>An application for managing a SharePoint team site with images, reports, and, XML content.</introduction>
      <summary>This template includes variables</summary>
      <version_major>9</version_major>
      <version_minor>3</version_minor>
      <build_number>38</build_number>
   </template_info>
   <appendpoint_list>
      <appendpoint>
         <ipv46>10.111.111.1</ipv46>
         <port>80</port>
         <servicetype>HTTP</servicetype>
      </appendpoint>
   </appendpoint_list>
   <service_list>
      <service>
         <ip>10.102.29.5</ip>
         <port>80</port>
         <servicetype>HTTP</servicetype>
      </service>
      <service>
         .
         .
         .
      </service>
      .
      .
      .
   </service_list>
   <variable_list>
      <variable>
         <name>body_size</name>
         <description>Evaluation Scope</description>
         <value>10000</value>
      </variable>
      <variable>
         .
         .
         .
      </variable>
      .
      .
```

```
        .
    </variable_list>
</template_deployment>
<!--NeedCopy-->
```

Deleting a Template File

September 21, 2020

If you no longer need an application template and its configuration, you can delete it. When you delete a template, the template XML file that is stored in the application template directory gets deleted. When you delete a template file, you are prompted to confirm the deletion. Click **Yes** to confirm and delete the selected file from the directory.

To delete a template file from the application template directory by using the GUI:

1. Navigate to **AppExpert > Applications** and then click **Manage Template**. Select a file from **Template Files** tab page or **Deployment Files** tab page and click **Delete**.

NetScaler Gateway Applications

September 21, 2020

When you configure an AppExpert application to manage a web application through the Citrix® NetScaler® appliance, you also create a set of application units and configure a set of traffic optimization and security policies for each unit. The policies that you configure for each application unit (policies for features such as Compression, Caching, and Rewrite) evaluate traffic that is meant only for that unit. In addition to these policies, you might want to configure Access Gateway policies for the application as a whole to optimize the application traffic when accessed through the Access Gateway. The Access Gateway Applications feature enables you to configure Access Gateway policies (Authorization, Traffic, Clientless Access, and TCP Compression) for an AppExpert application. After you configure NetScaler Gateway policies for AppExpert applications, you can include the policy configuration in the AppExpert application templates that you create.

You can also configure NetScaler Gateway policies for intranet subnets, file shares, and other network resources. Finally, you can create bookmarks for AppExpert applications and certain resources if you want users to be able to access them from the NetScaler Gateway home page.

You can configure the entities in the NetScaler Gateway Applications feature only by using the GUI.

How an NetScaler Gateway Application Works

When you create an AppExpert application in the Applications node in the GUI, a corresponding Access Gateway application is automatically created in the Access Gateway Applications node. Additionally, a rule that uses the AppExpert application's configured public endpoint is automatically created for the Access Gateway application entry. If multiple endpoints are configured for the AppExpert application, the rule includes all the configured public endpoints. The NetScaler appliance uses this rule to apply any configured Access Gateway policies to the traffic received at the AppExpert application's public endpoint. Traffic received at the AppExpert application's public endpoint is first evaluated against the NetScaler Gateway policies and then evaluated against the policies configured for AppExpert application's application units.

The rule that is created for the Clientless Access policies for an Access Gateway application is an advanced expression that also uses the public endpoint that is configured for the AppExpert application. Therefore, before you configure NetScaler Gateway policies for an AppExpert application, you must configure public endpoints for the AppExpert application.

When you include the NetScaler Gateway configuration in an application template, deployment-specific information, such as IP address and port information, and the rule that is created from this information are not included in the template.

How a NetScaler Configuration for a File Share Works

On the NetScaler appliance, you can configure Authorization policies for a file share that is hosted on your organization's network.

When you create a file share, you specify a name for the file share and the network path to the file share. In the network path, you can specify either the name of the server or the server IP address. A rule that uses the components of the file share path is automatically created for the file share. This rule enables the appliance to identify requests for files hosted on the file share server. Any Authorization policies that are configured for the file share are applied to incoming requests.

The NetScaler configuration for a file share cannot be saved in AppExpert application templates.

How a NetScaler Configuration for an Intranet Subnet Works

For the intranet subnets that form a part of your network, you can configure policies for Authorization, Traffic, and TCP Compression on the NetScaler appliance. When adding an intranet subnet, you specify the IP address and the netmask of the intranet subnet. A rule that uses these two parameters is automatically created for the intranet subnet. The appliance applies the configured policies to any request that has a destination IP address and netmask set to the subnet's IP address and netmask, respectively.

The NetScaler configuration for an intranet subnet cannot be saved in AppExpert application templates.

How the Other Resources Category Works

The Other Resources category enables you to configure Access Gateway policies for any network resource by using a rule of your choice. When you configure the NetScaler appliance to process requests for the network resource, you configure a classic expression to identify the requests that are associated with the network resource. You can configure Authorization, Traffic, Clientless Access, and TCP Compression policies for a network resource in Other Resources. The NetScaler appliance applies the configured NetScaler Gateway policies to any requests that match the configured rule.

The NetScaler configuration for a network resource in Other Resources cannot be saved in AppExpert application templates.

Entity Naming Conventions

The NetScaler Gateway Applications feature enforces a naming convention for some of the entities that you create in this feature. For example, the names of the profiles that you create for Traffic policies for an intranet subnet always begin with a string that consists of the name of the intranet subnet followed by an underscore (_). The name that you provide for the entity is appended to this string. If the name of a subnet is "subnet1," the name of the profile begins with "subnet1_." When such a naming convention is required (in the text box in which you type the name of an entity, for example), the user interface automatically inserts the string with which the name of the entity must begin and does not allow you to modify it.

Adding Intranet Subnets

September 21, 2020

You can specify authorization and Traffic policies for traffic that is bound for the intranet subnets that are configured in your network. The rules for these policies are automatically created by using the parameters you specify for the subnet.

To configure an intranet subnet by using the GUI:

1. In the navigation pane of the GUI, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, do one of the following:
    - To add an intranet subnet, click Intranet Subnets, and then click Add.
    - To modify an intranet subnet, click an intranet subnet, and then click Open.

3. In the Create Intranet Subnet or Configure Intranet Subnet dialog box, do the following:
    a) In the Name box, type a name for the intranet subnet you are adding. This parameter cannot be changed for an existing intranet subnet.
    b) In the IP Address box, type the IP address of the intranet subnet.
    c) In the Netmask box, type the netmask that will be used for the intranet subnet.
    d) Click Create or OK, and then click Close.

Adding Other Resources

September 21, 2020

For a network resource that you add to Other Resources, you must configure a classic expression that identifies the subset of traffic associated with the resource. For more information about configuring a classic expression, see the .

To configure a resource in Other Resources by using the GUI:

1. In the navigation pane of the GUI, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, do one of the following:
    - To add a resource, click Other Resources, and then click Add.
    - To modify a resource, click a resource, and then click Open.
3. In the Create Resource or Configure Resource dialog box, do the following:
    a) In the Name box, type a name for the resource you are adding. This parameter cannot be changed for an existing resource.
    b) In the Rule box, type the rule that will identify the subset of traffic that is associated with the resource you are adding.
    Alternatively, click Configure, and then create the rule in the Create Expression dialog box.
    c) Click Create or OK, and then click Close.

Configuring Authorization Policies

September 21, 2020

You can configure NetScaler Gateway authorization policies for AAA users and groups to access a resource.

To configure permissions for a AAA user or group to access a resource by using the GUI:

1. In the navigation pane of the GUI, expand AppExpert, and then click **Access Gateway Applications**.

2. In the details pane, in the Authorization column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure authorization policies for AAA users and groups.
3. Do one of the following:
    - If the AAA user or group for which you want to configure permissions is already in the Groups/Users tree, drag the user or group from the Groups/Users tree to the Users or Groups node in the `<application name>` tree. Then, right-click the user or group and click **Allow**.
    - If the AAA user or group for which you want to configure permissions is not configured on the appliance, in the `<application name>` tree, right-click Users or Groups, and then click Add. In the Create AAA Group or Create AAA User dialog box, fill in the values, click Create, and then click Close.
      The user or group is created with the permission set to Allow. To change the permission setting, right-click the group or user, and then click the permission setting.
4. Click **Close**.

Configuring Traffic Policies

September 21, 2020

The traffic policies that you configure for the resources in the NetScaler Gateway Applications node control client connections to the application. You do not have to configure a rule for the resource. The rule created automatically when you create the resource. You only need to associate a request profile with the traffic policy. In the traffic profile, you specify parameters such as the protocol, application time-out, and file type association.

To configure traffic policies for a resource

1. In the navigation pane of the GUI, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the Traffic column, click the icon provided for the application, file share, intranet subnet, or resource for which you want to configure traffic policies.
3. In the **Configure Traffic Policies** dialog box, do the following:
    - To specify an existing traffic policy, click **Insert Policy**, and then, in the Policy Name column, click the name of the policy.
    - To configure a new policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create Traffic Policy dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Request Profile, either select an existing request profile or click New to configure a new request profile. You can also select an existing profile and then click Modify to modify the profile.

  For more information about configuring a traffic policy or profile, see NetScaler Gateway.
  - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click Modify Policy. To modify only the associated profile, in the Profile column, click the name of the profile, and then click **Modify Profile**.
  - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.
  - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
  - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring Clientless Access Policies

September 21, 2020

Clientless access, when configured for a resource on the NetScaler appliance, allows end-users to access the resource without using the NetScaler Gateway client software. Users can use web browsers to access resources such as Outlook Web Access. You configure clientless access for a resource by configuring a clientless access policy that is associated with a clientless access profile.

To configure a clientless access policy for a resource in the NetScaler Gateway Applications node:

1. In the navigation pane of the GUI, expand **AppExpert**, and then click **Access Gateway Applications**.
2. In the details pane, in the **Clientless Access **column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a clientless access policy.
3. In the **Configure Clientless Access Policies** dialog box, do the following:
   - To specify an existing clientless access policy, click **Insert Policy**, and then, in the **Policy Name** column, click the name of the policy.
   - To configure a new clientless access policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create Clientless Access Policy dialog box, in the Name box, after the underscore (_), type a name for the policy. Then, in Profile, either select an existing profile or click New to configure a new profile. You can also select an existing profile and then click Modify to modify the profile.
     For more information about configuring a clientless access policy or profile, see NetScaler Gateway.
   - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click **Modify Policy**. To modify only the associated profile, in the Profile column, click the name of the profile, and then click Modify Profile.
   - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.

    - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes**, and then click **Close**.

Configuring TCP Compression Policies

September 21, 2020

You can configure TCP compression policies for an application to increase the performance of the application. TCP compression reduces network latency, reduces bandwidth requirements, and increases the speed of transmission. When configuring a TCP compression policy, you associate a compression action with the policy. The compression action specifies either Compress, GZIP, Deflate, or NoCompress as the compression type. For more information about the compression policies, and compression actions, see NetScaler Gateway.

To configure a TCP compression policy for a resource in the NetScaler Gateway Applications node

1. In the navigation pane of the GUI, expand AppExpert, and then click Access Gateway Applications.
2. In the details pane, in the TCP Compression column, click the icon for the application, file share, intranet subnet, or resource for which you want to configure a TCP compression policy.
3. In the Configure TCP Compression Policies dialog box, do the following:
    - To specify an existing TCP compression policy, click **Insert Policy**, and then, in the **Policy Name **column, click the name of the policy.
    - To create a new TCP compression policy, click Insert Policy, and then, in the Policy Name column, click New Policy. In the Create TCP Compression Policy dialog box, in the Policy Name box, after the underscore ("_"), type a name for the policy. Then, in Action, either select an existing action or click New and configure a new action. You can also click View to view the configured compression type.
      For more information about configuring a TCP compression policy or action, see NetScaler Gateway , Enterprise Edition at NetScaler Gateway.
    - To modify a policy that you have inserted, in the Policy Name column, click the policy name, and then click **Modify Policy**.
    - To regenerate the priorities assigned to the policies, click **Regenerate Priorities**.
    - To specify a new priority value for a policy, in the Priority column, double-click the assigned priority, and then enter the value you want.
    - To unbind a policy, click the policy, and then click **Unbind Policy**.
4. Click **Apply Changes,** and then click **Close**.

Configure bookmarks

October 23, 2020

You can configure bookmarks for internal applications or resources that are available for an entitled user. You can then bind the bookmark to a user, user group or virtual server globally and enable it for the user in the Access Interface. The bookmark links that you create appear on the websites panes under enterprise websites.

For more information, see Creating and Applying Web Links topic.

Customizing the configuration

September 21, 2020

After you verify that the AppExpert application is working correctly, you can customize the configuration to suit your requirements.

After you verify that the AppExpert application configuration is working correctly, you can configure the application and the deployment settings to suit your requirements. When you import an application template and deployment file, the system automatically populates the target application with the available configuration settings (such as application units, application unit rules, policies, persistence settings, load balancing methods, profiles, and traffic settings). In this application, you can configure deployment settings such as public endpoints, services, and service groups for each traffic subset. If you want the AppExpert application to manage a traffic subset that is not included in the template, you can either add an application unit for a traffic subset or modify the existing application unit. After you customize the configuration, you can also specify the order of evaluation for each traffic subset that the application manages.

Configuring an AppExpert application consists of the following steps:

1. Configuring Public Endpoints
2. Configuring Application Units
3. Specifying the Order of Evaluation
4. Viewing Application Configuration using Visualizer

Also, you can configure the policies that the template provided. If the AppExpert application template does not include policies for a particular NetScaler feature, such as Rewrite or application firewall, you can configure your own policies.

Configure public endpoints

September 21, 2020

If you did not specify a public endpoint when importing an AppExpert application, you can specify public endpoints after you create the application. You can configure one public endpoint of type HTTP and one public endpoint of type HTTPS for your AppExpert application.

If endpoints are already configured for the application, you can dissociate endpoints from the AppExpert application and delete any endpoints that you no longer need. Note that when you dissociate a public endpoint from the AppExpert application, the endpoint is automatically unbound from the associated application unit, but it is not deleted from the system.

To configure public endpoints for an AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to configure public endpoints, and then click Edit.
3. In the **Applications** page, go to **Public Endpoint** section and click the pencil icon.
4. In the **Public Endpoint** slider, set the following parameters.
    a) Public endpoint type. Select the radio button to define the endpoint type.
    b) Name. Name of the public endpoint.
    c) IP address. IP address of the public endpoint.
    d) Port. Port number of the public endpoint.
    e) Protocol. Select a protocol type as HTTP or HTTPS.
5. Click **Continue.**
6. In the **Application Units** section, select an application unit from the list.
7. Click **Continue** to set the policy and server details.
8. Click **OK** and then Done.
9. Click Close.

For more information about the parameters in the **Configure Public Endpoint** dialog box, see Content Switching.

Configuring Endpoints for an Application Unit

September 21, 2020

When you configure multiple public endpoints for an AppExpert application, by default, all endpoints are bound to each application unit, and each application unit processes the requests received by all the endpoints. However, you can specify that a given application unit manages the traffic that is received by only a subset of the endpoints that are configured for the AppExpert application.

To configure endpoints for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to specify public endpoints, and then click Configure Public Endpoints.
3. In the Choose Public Endpoints dialog box for the application unit, do one of the following:
   - If you are specifying endpoints for the application unit for the first time, clear the check boxes that correspond to the endpoints that you do not want to be bound to the application unit.
   - If you want to specify endpoints that are listed in the dialog box but not currently bound to the application unit, click the corresponding check boxes.
4. Click **OK**.

Configure services and service groups for an application unit

September 21, 2020

When you configure a service or service group, you either modify an existing service or service group, or add new services to the AppExpert application. You add services or service groups if you did not specify them when you imported the application template. You also add services and service groups when you increase the number of servers that host instances of the application. You can configure a service and service group for an application unit only after you configure the service or service group for the AppExpert application.

To configure a service or service group for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application and then click **Edit.**
3. In the **Applications** page, select an application unit and then click **Continue**.
4. In the **Services and Service Groups** section, do the following:
   a) In the Service Binding slider, set the following parameters.
      i. Service. Select a load balancing service from the list or create a new service.
      ii. Weight. Provide a weight value for the service.
   b) Click **Bind** and then **Done.**
   c) In the ServiceGroup Binding slider, set the following parameters:
      i. Service Group Name. Select a load balancing service group or create a new service group.
      ii. Click **Bind** and then **Done**.

    d) Click **Done**.
5. Click **Continue** to set other configurations.

Configuring services, service groups, and load balancing Parameters for an application unit

September 21, 2020

When you configure services and service groups for an AppExpert application, by default, all the services and service groups are bound to each application unit. However, depending on how you have configured your web application, the application resources that are managed by an application unit might be hosted on only some of the servers that are configured as services for the AppExpert application. Or, a set of servers might host content that is meant for the requests received at one or more specific public endpoints. In such scenarios, if all the services and service groups that are configured for the AppExpert application are associated with the application unit, a request that is forwarded to a server that does not host the requested content might not be served or might be served incorrect content. Therefore, you must ensure that each application unit is configured to manage only those services that can serve the requested content.

When configuring services and service groups for an application unit, you might choose to specify load balancing settings such as the weights that services must be assigned and the desired load balancing, persistence, and spillover methods. For more information about these settings, see Load Balancing.

To configure services or service groups for an application unit

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application unit for which you want to configure a service or service group, and then click **Configure Backend Services**.
3. In the **Configure Backend Services** dialog box, do one of the following:
   - To configure services, click the **Services** tab.
   - To configure service groups, click the **Service Groups** tab.
4. In the **Services** or **Service Groups **tab, do one of the following:
   - Clear the check boxes that correspond to the services or service groups that you do not want configured for the application unit. Make sure that the check boxes that correspond to the services or service groups that you want configured for the application unit are selected. Then, in the Weight column, specify the weight that you want to assign to each configured service.
   - To specify all services or service groups, click **Activate All**.
5. On the **Method** and **Persistence** and **Advanced** tabs, specify the desired parameters.
6. Click **OK**.

Create application units

September 21, 2020

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications** page, go to **Application Units** section and click the **pencil** icon.

To configure policy expressions for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications **page, go to** Application Units** section and click the **+ **icon. to create a unit and add policy expressions.
4. To specify the format of the new expression, do one of the following:
    a) To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
    b) To specify that you want to configure an advanced expression in the Rule box, click Default Syntax.
    c) In the Rule box, configure the expression.
5. Click **OK**.

Configuring application unit rules

September 21, 2020

You might want to configure an application unit rule to include or exclude certain types of traffic. When you configure the rule, you can also define the syntax of the expression.

To configure an application unit rule:

1. In the navigation pane of the GUI, expand **AppExpert**, and then click **Applications**.
2. In the details pane, right-click the application unit for which you want to modify the rule, and then click **Open**.

3. In the **Configure Application Unit** dialog box, do the following:
    a) To specify the format of the new expression, do one of the following:
        - To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
        - To specify that you want to configure an advanced expression in the Rule box, click **Default Syntax**.
    b) In the Rule box, configure the expression.
4. Click **OK**.

Configuring policies for application units

September 21, 2020

For an AppExpert application, you can configure policies for Compression, Caching, Rewrite, Responder, and Application Firewall. The templates that you download from the Citrix Community web site provide you with a set of policies that fulfill the most common application management requirements. You might want to fine-tune or customize these policies. If the set of policies provided for a given application unit does not include policies for a particular feature, you can create and bind your own policies for that feature.

If you create an AppExpert application without using a template, you must configure all the policies that the web application needs.

The GUI uses various icons to indicate whether or not policies are configured for a feature. For an application unit, if a policy is configured for a given feature, an icon that represents the feature is displayed. For example, if a compression policy is configured for an application unit, a compression icon is displayed in the Compression column for the application unit. For features for which no policy is configured, an icon depicting a plus sign (+) is displayed.

> **Note**:
>
> When configuring policies for application units, you might need to configure policies and expressions that are either in the classic or
> default syntax. Additionally, when you configure
> default syntax policies, you might need to specify parameters such as Goto expressions and invoke policy banks.
> For information about configuring policies and expressions in both formats, see Policies and Expressions.

Configuring compression policies

You can use either classic policies or advanced policies to configure compression, but you cannot bind compression policies of both types to the same application unit.

To configure a compression policy for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Compression column.
3. In the Configure Compression Policies dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - Click Switch to Default Syntax if you want to configure a default syntax compression policy. If you want to bind or configure classic compression policies, and if you are in the default syntax view, you can click Switch to Classic Syntax to return to the classic policy view and begin modifying bound classic policies or create and bind new classic compression policies.
      Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you are in the
      default syntax view, when you click
      Insert Policy, the list that appears in the
      Policy Name column will include only
      default syntax policies. You cannot bind policies of both types to an application unit.
    - If you want to configure classic policies, click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
      You can configure both request-time and response-time classic compression policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
    - To modify a compression policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Compression Policy dialog box, modify the policy, and then click OK.
      For information about modifying a compression policy, see Compression.
    - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
    - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
    - To regenerate assigned priorities, click **Regenerate Priorities**.
    - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Compression Policy** dialog box, configure the policy, and then click **Create**.
      For information about modifying a compression policy, see Compression.
    - If you are configuring a default syntax expression, do the following:

    - In the **Goto Expression** column, select a Goto expression.
    - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Caching Policies

You can use only default syntax policies and expressions to configure Caching policies.

To configure Caching policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Caching column.
3. In the **Configure Cache Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
      You can configure both request-time and response-time Caching policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
    - To modify a Caching policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy.** Then, in the **Configure Cache Policy** dialog box, modify the policy, and then click **OK**.
      For information about modifying a Caching policy, see Integrated Caching.
    - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
    - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
    - To regenerate assigned priorities, click **Regenerate Priorities**.
    - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the Policy Name column, click **New Policy**. Then, in the **Create Cache Policy** dialog box, configure the policy, and then click **Create**.
      For information about modifying a Caching policy, see Integrated Caching.
    - In the Goto Expression column, select a Goto expression.
    - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Rewrite policies

You can use only default syntax policies and expressions to configure Rewrite policies.

To configure Rewrite policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Rewrite column.
3. In the **Configure Rewrite Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - Click either Request or Response, depending on whether you want the policy to be evaluated at request-time or at response-time.
      You can configure both request-time and response-time Rewrite policies for an application unit. After evaluating all of the request-time policies, if no match is found, the appliance evaluates response-time policies.
    - To modify a Rewrite policy that is already bound to the application unit, click the name of the policy, and then click Modify Policy. Then, in the Configure Rewrite Policy dialog box, modify the policy, and then click OK.
      For information about modifying a Rewrite policy, see Rewrite.
    - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
    - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
    - To regenerate assigned priorities, click **Regenerate Priorities**.
    - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Rewrite Policy** dialog box, configure the policy, and then click **Create**.
      For information about modifying a Rewrite policy, see Rewrite.
    - In the Goto Expression column, select a Goto expression.
    - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Responder policies

You can use only default syntax policies and expressions to configure Responder policies.

To configure Responder policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Responder column.
3. In the **Configure Responder Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
    - To modify a Filter policy that is already bound to the application unit, click the name of the policy, and then click **Modify Policy**. Then, in the **Configure Responder Policy** dialog

  box, modify the policy, and then click **OK**.
  For information about modifying a Responder policy, see Responder.
  - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
  - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
  - To regenerate assigned priorities, click **Regenerate Priorities.**
  - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name column**, click **New Policy**. Then, in the **Create Responder Policy** dialog box, configure the policy, and then click **Create**.
    For information about modifying a Responder policy, see Responder.
  - In the Goto Expression column, select a Goto expression.
  - In the Invoke column, specify the policy bank that you want to invoke if the current policy evaluates to TRUE.
4. Click **Apply Changes**, and then click **Close**.

Configuring Application Firewall policies

You can configure both classic and default syntax policies and expressions for Application Firewall. However, if a policy of one type is already bound globally or to a virtual server that is configured on the appliance, you cannot bind a policy of the other type to an application unit. For example, if a default syntax policy is already bound either globally or to a virtual server, you cannot bind a classic policy to an application unit.

To configure Application Firewall policies for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, in the row for the application unit you want to configure, click the icon provided in the Application Firewall column.
3. In the **Configure Application Firewall Policies** dialog box, do one or more of the following, depending on the configuration tasks you want to perform:
   - Click either Classic Expression or Advanced Expression depending on the type of expression you want to configure for the Application Firewall policy.
     Important: This setting also determines what policies are displayed when you want to insert a policy. For example, if you select
     Advanced Expression, when you click
     Insert Policy, the list that appears in the
     Policy Name column will include only
     default syntax policies. You cannot bind policies of both types to an application unit. This option is not available if a policy of either type is already bound either globally or to a virtual server.
   - To modify an application firewall policy that is already bound to the application unit, click

    the name of the policy, and then click Modify Policy. Then, in the Configure Application Firewall Policy dialog box, modify the policy, and then click OK.
    For information about modifying a application firewall policy, see Policies.
  - To unbind a policy, click the name of the policy, and then click **Unbind Policy**.
  - To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
  - To regenerate assigned priorities, click **Regenerate Priorities**.
  - To insert a new policy, click **Insert Policy** and, in the list that is displayed in the **Policy Name** column, click **New Policy**. Then, in the **Create Application Firewall Policy** dialog box, configure the policy, and then click **Create**.
    For information about modifying a application firewall policy, see Policies.
4. Click **Apply Changes**, and then click **Close**.

Create application units

September 21, 2020

You might need to add application units for traffic subsets that are either specific to your web application implementation or not defined in the template. When creating an application unit, you must configure a rule for the application unit.

To create an application unit for the AppExpert application:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications** page, go to **Application Units** section and click the **pencil** icon.

To configure policy expressions for an application unit:

1. Navigate to **AppExpert** > **Applications**.
2. In the details pane, right-click the application for which you want to add an application unit, and then click **Add**.
3. In the **Applications **page, go to** Application Units** section and click the **+ **icon. to create a unit and add policy expressions.
4. To specify the format of the new expression, do one of the following:
   a) To specify that you want to configure a classic expression in the Rule box, click Classic Syntax.
   b) To specify that you want to configure an advanced expression in the Rule box, click Default Syntax.
   c) In the Rule box, configure the expression.

5. Click **OK**.

Configuring public endpoints for an application

September 21, 2020

To configure public endpoints for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and then click **Edit**.
2. In the **Public Endpoint** section, click **+** to configure a new public endpoint.
3. In the **Public Endpoint** slider, do one of the following:
    a) Click **New** to create a new endpoint.
    b) Click **Existing Public Endpoint** to select an endpoint from the drop-down list.
4. Set the following endpoint parameters:
    a) Name
    b) IP address
    c) Protocol
    d) Port
5. Click **Continue** to configure additional settings such as application units, GSLB server bindings, policies, profiles, push, traffic settings, and authentication.
6. Click **OK** and then **Done**.
7. Click **Continue** and then **Done**.

To edit a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications**, select an application, and click **Edit**. In the **Public Endpoint** section, select an endpoint, click the pen icon, and modify the endpoint settings.

To delete a public endpoint for an application by using the GUI:

Navigate to **AppExpert** > **Applications** > **Public Endpoint,** click the pen icon to view the delete icon next to the entity.

NetScaler's video tutorials enable you to understand NetScaler features in an easy and simply way. Watch <`https://www.youtube.com/watch?v=z4v-edQiVpw`> video to learn how to configure a public endpoint.

Specifying the Order of Evaluation of Application Units

September 21, 2020

Application unit rules are evaluated in the order in which they are placed in the GUI. The rule that is configured for the topmost application unit is always configured first, followed by the rule that is configured for the second topmost application unit, and so on. The default application unit is always evaluated last.

When a request matches the rule that is configured for an application unit, the request is processed by the application unit, and no further matching is performed. Therefore, the order of evaluation of application units becomes an important factor if the traffic subsets for two or more application units overlap. If the traffic subsets for two or more application units overlap, you must specify the order in which an incoming request is matched against the application unit rules.

To specify the order of evaluation of application units:

1. Navigate to **AppExpert** > **Applications**, select an application and click **Edit**. In the **Application Unit** section, click the **Pencil** icon and then hover the cursor over the check box to the left of the name of the application unit. Click the icon that appears next to the check box and hold down the mouse to drag the application up or down to a new location in the priority list.

Configuring Persistency Groups for Application Units

September 21, 2020

You can configure a persistency group for the application units in an AppExpert application. In the context of an AppExpert application, a persistency group is a group of application units that you can treat as a single entity for the purpose of applying common persistence settings. When the application is exported to an application template file, the persistency group settings are included, and they are automatically applied to the application units when you import the AppExpert application.

To configure a persistency group for an application by using the GUI:

1. Navigate to **AppExpert** > **Applications**.
2. In the **Applications View** dialog box, click the name of the application for whose application units you want to configure a persistency group, and then click **Configure Persistency Groups**.
3. In the **Configure Persistency Groups** dialog box, do one of the following:
   - To add a persistency group, click **Add**.
   - To modify a persistency group, click **Open**.
4. In the **Create Persistency Group** or **Configure Persistency Group** dialog box, set the following parameters:
   - Group Name—Name of the persistency group. For the NetScaler appliance to recognize the persistency group as part of the application's configuration, the name of the AppExpert application must be included in the name of the persistency group, as a prefix. Therefore, by default, the appliance displays the prefix in the Group Name box, and you cannot

remove that prefix. Enter a name of your choice after the prefix.
   - Persistence—Type of persistence for the virtual server. If you select SOURCEIP, in the IPv4 Netmask box, enter a network mask that specifies the number of bits that the appliance must consider when creating persistence sessions. If you select COOKIEINSERT, in the Cookie Domain and Cookie Name boxes, specify a domain attribute to send in the Set-Cookie directive, and a name for the cookie, respectively.
   - Timeout—Time period for which a persistence session is in effect.
   - Backup Persistence—Type of backup persistence for the group.
   - Backup Timeout—Time period, in minutes, for which backup persistence is in effect.
   - Application Units—To add an application unit to the persistency group, in the Available Application Units box, click the application unit, and then click **Add**. To remove an application unit from the persistency group, in the Configured Application Units box, click the application unit, and then click **Remove**.
5. Click **OK**.

Viewing AppExpert Applications and Configuring Entities by Using the Application Visualizer

September 21, 2020

The Visualizer feature shows you a graphical representation of an application's configuration. It includes the name of the public endpoint, application units assigned to the public endpoint, and the number of policies and services bound to the application. You can use the Visualizer to obtain a visual overview of an AppExpert application's configuration and configure some of the displayed entities. By default, the Visualizer displays application units, services, and monitors for the selected application.

To view an AppExpert application by using the Application Visualizer:

1. Navigate to **AppExpert** > **Applications**, select an application entity, and click **Visualizer**.

AppQoE

September 21, 2020

Application level Quality of Experience (AppQoE) integrates several existing policy-based security features of the NetScaler appliance into a single integrated feature that takes advantage of a new queuing mechanism, fair queuing. Fair queuing manages requests to load-balanced web servers and applications at the virtual server level instead of at the service level, allowing it to handle queuing of all requests to a web site or application as one group before load balancing, instead of as separate streams after load balancing.

The features that are integrated into AppQoE are HTTP Denial-of-Service Protection (HDOSP), Priority Queuing (PQ), and SureConnect. Collectively these services provide protection against a number of problems:

- **Simple overload.** Any server, no matter how robust, can accept only a limited number of connections at one time. When a protected web site or application receives too many requests at once, the Surge Protection feature detects the overload and queues the excess connections til the server can accept them. The Priority Queuing feature ensures that whoever most needs access to a resource is provided access without having to wait behind other lower-priority requests. The SureConnect feature displays an alternate web page that notifies users that the resource that they requested is not available.
- **Denial-of-Service (DOS) attacks.** Any public-facing resource is vulnerable to attacks whose purpose is to bring that service down and deny legitimate users access to it. The Surge Protection, Priority Queuing, and SureConnect features help manage DOS attacks as well as other types of high load. In addition, the HTTP Denial-of-Service Protection feature targets DOS attacks against your web sites, sending challenges to suspected attackers and dropping connections if the clients do not send an appropriate response.

Until the current version of the NetScaler operating system, these features were implemented at the service level, which means that each service was assigned its own queues. While service-level queues work, they also have some disadvantages, most of which are due to the NetScaler appliance having to load balance requests before implementing any of the protection features that rely on queuing. Implementing protection features before queuing has a number of advantages, some of which are listed below:

- Absolute priority of connections as configured in the priority queuing feature can be maintained.
- Connections are not flushed if a service transitions state, as they are in a service-level queue.
- During periods of high load, such as a denial-of-service attack, HTTP DoS and SureConnect come into play before load balancing, allowing these features to detect and divert unwanted or lower-priority traffic from the load balancer before the load balancer must cope with it.

In addition to implementing fair queuing, AppQoE integrates a set of features that each provide a different set of tools to achieve a common goal: protecting your networked resources from excessive or inappropriate demand. Putting these features into a common framework enables you to configure and implement them more easily.

Enabling AppQoE

September 21, 2020

To configure AppQoE, you must first enable the feature.

To enable AppQoE by using the command line

At the command prompt, type the following commands:

- enable ns feature appqoe
- show ns feature

Example:

```
> enable ns feature appqoe
 Done
> show ns feature

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

To enable AppQoE by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features**.
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

AppQOE actions

September 21, 2020

After enabling the AppQoE feature, you must configure one or more actions for handling requests.

> **Important**: No specific individual parameters are required to create an action, but you must include at least one parameter or you cannot create the action.

To configure an AppQoE action by using the command line

At the command prompt, type the following commands:

- `add appqoe action <name> [-priority <priority>] [-respondWith (ACS|NS)[<customfile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction (**SimpleResponse** | **HICResponse**)]`
- `show appqoe action`

Example

To configure priority queuing with policy queue depths of 10 and 1000 for medium and lowest priority queues, respectively:

```
> add appqoe action appqoe-act-basic-prhigh -priority HIGH
 Done

> add appqoe action appqoe-act-basic-prmedium -priority MEDIUM -
   polqDepth 10
 Done

> add appqoe action appqoe-act-basic-prlow -priority LOW -polqDepth
   1000
 Done

> show appqoe action
1)       Name: appqoe-act-basic-prhigh
         ActionType: PRIORITY_QUEUING
         Priority: HIGH
         PolicyQdepth: 0
         Qdepth: 0

2)       Name: appqoe-act-basic-prmedium
         ActionType: PRIORITY_QUEUING
         Priority: MEDIUM
         PolicyQdepth: 10
         Qdepth: 0

3)       Name: appqoe-act-basic-prlow
         ActionType: PRIORITY_QUEUING
```

```
        Priority: LOW
        PolicyQdepth: 1000
        Qdepth: 0
Done
<!--NeedCopy-->
```

To modify an existing AppQoE action by using the command line

At the command prompt, type the following commands:

- `set appqoe action <name> [-priority <priority>] [-altContentSvcName <string>] [-altContentPath <string>] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-maxConn <positive_integer>] [-delay <usecs>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse | HICResponse )]`
- `show appqoe action`

To remove an AppQoE action by using the command line

At the command prompt, type the following commands:

- `rm appqoe action <name>`
- `show appqoe action`

Parameters for configuring an AppQoE action

- name

  A name for the new action, or the name of the existing action that you want to modify. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.

- priority

  The priority queue to which the request is assigned. When a protected web server or application is heavily loaded and cannot accept additional requests, specifies the order in which waiting requests are to be fulfilled when resources are available. The choices are:

  1. **HIGH.** Fulfills the request as soon as resources are available.
  2. **MEDIUM.** Fulfills the request after it has fulfilled all requests in the HIGH priority queue.
  3. **LOW.** Fulfills the request after it has fulfilled all requests in the HIGH and MEDIUM priority queues.
  4. **LOWEST.** Fulfills the request only after it has fulfilled all requests in higher-priority queues.

If priority is not configured, then the NetScaler appliance assigns the request to the LOWEST priority queue by default.

- respondWith

  Configures the NetScaler to take the specified Responder action when the specified threshold is reached. Must be used with one of the following settings:

  - **ACS:** Serves content from an alternate content service. Threshold: maxConn (maximum connections) or delay.
  - **NS:** Serves a built-in response from the NetScaler. Threshold: maxConn (maximum connections) or delay.
  - **NO ACTION:** Serves no alternative content. Assigns connections to the LOWEST priority queue if the maxConn (maximum connections) or delay threshold is reached.

- altContentSvcName

  If
  -responseWith ACS is specified, the name of the alternative content service, usually an absolute URL to the web server that hosts the alternate content.

- altContentPath

  If
  -responseWith (ACS | NS) is specified, the path to the alternative content.

- polqDepth

  Policy queue depth threshold value for the policy queue associated with this action. When the number of connections in the policy queue associated with this action increases to the specified number, subsequent requests are assigned to the LOWEST policy queue. Minimum value: 1 Maximum value: 4,294,967,294

- priqDepth

  Policy queue depth threshold value for the specified priority queue. If the number of requests in the specified queue on the virtual server to which the policy associated with the current action is bound increases to the specified number, subsequent requests are assigned to the LOWEST priority queue. Minimum value: 1 Maximum value: 4,294,967,294

- maxConn

  The maximum number of connections that can be open for requests that match the policy rule. Minimum value: 1 Maximum value: 4,294,967,294

- delay

  The delay threshold, in microseconds, for requests that match the policy rule. If a matching request has been delayed for longer than the threshold, the

NetScaler appliance performs the specified action. If NO ACTION is specified, then the appliance assigns requests to the LOWEST priority queue. Minimum value: 1 Maximum value: 599999,999

- dosTrigExpression

  Adds an optional second-level check to trigger DoS actions.

- dosAction

  Action to take when the ADC determines that it or a protected server is under DoS attack. Possible values: SimpleResponse, HICResponse.

These values specify HTTP challenge-response methods for validating the authenticity of incoming requests to mitigate an HTTP-DDoS attack.

In the HTTP challenge-response generation and validation process, AppQoE uses cookies to validate the client's response and verify that the client seems to be genuine. When sending a challenge, a NetScaler appliance generates two cookies:

Header cookie (_DOSQ). Contains client-specific information, so that the NetScaler appliance can verify the response.

Body cookie (_DOSH). Information used to validate the client machine. The client's browser (or the user, in the case of HIC) computes a value for this cookie. The NetScaler appliance compares that value with the expected value to verify the client.

The information that the appliance sends to the client for computing the _DOSH value is based on the DoS Action configuration.

1. SimpleResponse: In this case, a NetScaler appliance splits the value and generates a JavaScript code to combine the final value. A client machine capable of computing the original value is considered genuine.
2. HICResponse: in this case, a NetScaler appliance generates two single-digit numbers and generates images for those numbers. Then, using a backpatch framework, the appliance inserts those images as base64 strings.

Limitations:

1. This is not a trivial CAPTCHA implementation, which is why that term not used.
2. The validation number is based on a NetScaler-generated number that does not change for 120s. This number should be dynamic or client specific.

To configure an AppQoE action by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
    - To create a new action, click **Add**.

  - To modify an existing action, select the action, and then click **Edit**.
3. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):
   - Name—name
     > - Action type—respondWith
   - Priority—priority
   - Policy Queue Depth—polqDepth
   - Queue Depth—priqDepth
   - DOS Action—dosAction
4. Click **Create** or **OK**.

AppQoE parameters

September 21, 2020

In the AppQoE parameters, you configure the session life of an AppQoE session, the file name of the file containing the customized response, and the number of client connections that can be placed in a queue.

To configure the AppQoE parameter settings by using the command line

At the command prompt, type the following commands:

- `set appqoe parameter [-sessionLife <secs>] [-avgwaitingclient <positive_integer>] [-MaxAltRespBandWidth <positive_integer>] [-dosAttackThresh <positive_integer>]`
- `show appqoe parameter`

Parameters for configuring the AppQoE parameters

- sessionLife

  Number of seconds to wait after displaying alternate content before the ADC displays the same content again. Default value: 300 Maximum Minimum value: 1 Maximum value: 4,294,967,294

- avgwaitingclient

  The average number of client requests that can be in the service waiting queue. Default value: 1000000 Maximum value: 4,294,967,294

- MaxAltRespBandWidth

  The maximum bandwidth to consume when sending alternate responses. If the maximum is reached, the ADC quits sending the alternate content til bandwidth consumption drops. Default value: 100 Minimum value: 1 Maximum value: 4,294,967,294

- dosAtckThrsh

  The denial-of-service attack threshold. The number of connections that must be waiting in queues before the ADC responds with DoS protection measures. Default value: 2000 Minimum value: 0 Maximum value: 4,294,967,294

To configure the AppQoE parameter settings by using the GUI

1. Navigate to **AppExpert** > **AppQoE**.
2. In the details pane, click **Configure AppQoE Parameters**.
3. In the **Configure AppQoE params** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Parameters" as follows (asterisk indicates a required parameter):
   - Session Life (secs)
   - sessionLife
   - Average waiting client—avgwaitingclient
   - Alternate Response Bandwidth Limit(Mbps) —MaxAltRespBandWidth
   - DOS Attack Threshold —dosAttackThresh
4. Click **OK**.

AppQoE Policies

September 21, 2020

To implement AppQoE, you must configure at least one policy to tell your NetScaler how to distinguish the connections to be queued in a specific queue.

To configure an AppQoE policy by using the command line

At the command prompt, type the following command:

```
add appqoe policy <name> -rule <expression> -action <string>
```

Example:

The following example selects requests with a User-Agent header that contains "Android", and assigns them to the medium priority queue. These requests come from smartphones and tablets that run the Google Android operating system.

```
> add appqoe action appqoe-act-primd -priority MEDIUM
Done
> add appqoe policy appqoe-pol-primd -rule "HTTP.REQ.HEADER("User-Agent
   ").CONTAINS("Android")" -action appqoe-act-primd
Done
> sh appqoe policy appqoe-pol-primd
        Name: appqoe-pol-primd
        Rule: HTTP.REQ.HEADER("User-Agent").CONTAINS("Android")
        Action: appqoe-act-primd
        Hits: 0

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

Parameters for configuring an AppQoE policy

- name. A name for the AppQoE policy. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the type of action.
- rule. A NetScaler expression that tells the appliance which connections it should handle.
- action. The AppQoE action to perform when a connection matches the policy.

To configure an AppQoE policy by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Policies**.
2. In the details pane, do one of the following:
   - To create a new policy, click **Add**.
   - To modify an existing policy, select the policy, and then click **Edit**.
3. If you are creating a new policy, in the **Create AppQoE Policy** dialog, in the Name text box, type a name for your new policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols. You should chose a name that helps identify the purpose and effect of this policy.

If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.

4. In the **Action** drop-down list, choose the AppQoE action to perform when the policy matches a connection. Click the plus (+) to open the **Add AppQoE Action** dialog and add a new action.
5. In the **Rule** text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:
    a) In the **Create Expression** dialog box, click **Add**.
    b) In the Add Expression dialog box, select a common expression from the Frequently Used Expressions drop-down list, or use the Construct Expression drop-down lists to create the expression that defines which traffic to filter.

       If you choose to create your own expression, you start by selecting the first term from the first drop-down list on the left side of the Construct Expression area. The choices in that list are:

       - HTTP: All traffic to port 80 and port 443.
       - SYS:
       - CLIENT:
       - SERVER:
       - ANALYTICS:
       - TEXT:

       The default choice is HTTP. After you make a choice in the first drop-down list (or accept the default), you can choose the next term in your expression from the drop-down list to the right of it. The terms in that list and other lists that follow change depending on your previous choices; the lists offer only terms that are valid choices. Continue to select terms until you have finished the expression.
    c) When you have created the expression that you want, click **OK**. The expression is added in the **Expression** text box.
6. Click **Create**. The expression appears in the **Rule** text box.

Entity templates

September 21, 2020

An entity template is a collection of configuration information for an individual entity on a NetScaler® appliance. It provides a specification and a set of defaults for a configurable NetScaler entity, such as a policy, virtual server, service, or action. By using a template that defines a set of defaults, you can quickly configure multiple entities that require a similar configuration while eliminating several configuration steps.

Entity templates are available only in the GUI. You use the NetScaler GUI to create, manage, and use any type of entity template. You can share entity templates with other administrators and manage local folders that contain the templates. You can also import entity templates from and export entity templates to your local computer.

Before creating a template, you should be familiar with the configuration of the entity.

Note: You use entity templates to configure individual entities. To configure multiple entities related to a particular Web application, you must use an application template. For more information, see “ AppExpert Applications and Templates.”

How Entity Templates Work:

When you create a template for a NetScaler entity, you specify default values for the entity. You specify what values must be read-only, what values must not be displayed, and what values users can configure. You also configure the pages that compose the template import wizard. All the information and settings you provide are stored in the template file.

When a user imports the entity template to a NetScaler appliance, a wizard guides the user through the various pages that you configured for the template. The wizard displays the read-only parameter values and prompts the user to specify values for the configurable parameters. After the user follows the instructions in the wizard, the appliance creates the entity with the configured values.

For example, you can create an entity template for HTTP services that provides a text box for a service name and assigns preset values for the service protocol, timeouts, thresholds, and monitors. Later, when you use the template to create new HTTP services, a wizard prompts you for a service name and supplies the preset values that you would otherwise have configured manually.

The procedure for creating entity templates for load balancing virtual servers is different than the AppExpert procedure for creating other entity templates. For more information, see “Creating an Entity Template.”

In addition, the procedure for using the template to create the load balancing virtual server entity is different. For more information, see “Creating an Entity from a Template.”

Configuring an entity template

September 21, 2020

You can create or modify an entity template either from the AppExpert feature node or from the associated NetScaler feature node for the entity. For example, you can create a content switching virtual server entity template in either the AppExpert feature node or the content switching feature node in the GUI.

If you create a template that is not based on an existing entity, you can specify the following options and settings for the template:

- The default value of a parameter.
- Whether the default values are visible to users.
- Whether the default values can be changed by users.
- The number of pages in the entity import wizard, including the page names, text, and available parameters.
- The entities that must be bound to the entity for which the template is being created.

  For example, when you are creating a cache redirection virtual server template, you can specify the policies that you want to bind to the cache redirection virtual servers that you create from the template. However, only binding information is included in the template. The bound entities are not included. If the entity template is imported to another NetScaler appliance, the bound entities must exist on the appliance at import time for the binding to succeed. If none of the bound entities exist on the target appliance, the entity (for which the template was configured) is created without any bindings. If only a subset of the bound entities exist on the target appliance, they are bound to the entity that is created from the template.

When you create a template based on an existing entity, the configuration settings of the entity appear in the template. All bound entities are selected by default, but you can modify bindings as necessary. As in the case of a template that is not based on an existing entity, only binding information is included and not the entities. You can either save the template with the existing configuration settings or use the settings as a basis for creating a new configuration for a template.

Configuring variables in load balancing virtual server templates

September 21, 2020

Load balancing virtual server templates support the declaration of variables in the configured load balancing parameters and in bound policies and actions. The ability to declare variables enables you to replace preconfigured values with values that suit the environment into which you are importing the template. The Entity Template Wizard, which appears when you import a template, includes a Specify Variable Values page on which you can specify appropriate values for the variables that are configured for the entity template. This wizard page appears only when you import a template that is configured with existing variables.

As an example, consider the following expression configured for a policy that is bound to a load balancing virtual server for which you are creating a template. The expression evaluates the value of the Accept-Language header in an HTTP request.

```
HTTP.REQ.HEADER("Accept-Language").CONTAINS("en-us")
```

If you want the value of the header to be configurable at import time, you can specify the string en-us as a variable. When importing the template, you can specify a new value for the variable on the Specify Variable Values page.

After you create a variable, you can do the following:

- Assign additional strings to an existing variable. After you create a variable for a string, you can select and assign other parts of the same or different expression to the variable. The strings you assign to a variable need not be the same. At import time, all the strings that are assigned to the variable are replaced with the value that you provide.
- View the string or strings that are assigned to the variable.
- View a list of all the entities and parameters that use the variable.

To configure variables in a load balancing virtual server template

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**
2. In the details pane, right-click the virtual server that you want to export to a template file, and then click **Create Template**.
3. In the **Create Template** dialog box, modify the default template file name if required, specify the location where you want to save the template, and then click **Configure Variables**.
4. In the **Configure Variables** dialog box, click the tab that lists the entity for which you want to configure a variable, select the entity, and then click **Configure Variables**.
5. In the Variables for Entity Type: Entity Name dialog box, click the √x button next to the parameter value or expression in which you want to create a variable.
6. In the variables for Field Name dialog box, do the following:
    - To create a variable, in the text box that displays the configured expression or value, select the string that you want to be configurable at import time, and then click **Add**. In the **Create Variable** dialog box, specify a name and a description for the variable, and then click **Create**.

      The name of the variable, its value, and the description you provided appear in the Available Variables listing in the dialog box. The name you provide will be the name of the associated field in the template import wizard, and the description will appear as alt text when the user positions the mouse pointer over the field.

- To modify a variable, in the Available Variables list, click the variable, and then click **Open**. In the **Create Variable** dialog box, modify the value and the description, and then click **OK**.

  The new value that you specify will not replace the text selected in the text box that displays the configured expression or value. However, when you import the template, the new value will be displayed as the default value for the variable in the template import wizard.

- To view all the strings that are assigned to a given variable, in the Available Variables listing, click the name of the variable. The strings that are assigned to the variable are highlighted.
- To view a list of all the parameters, expressions, and actions in which the variable is used, in the Available Variables listing, click the variable whose references you want to view, and then click **Show References**.
- To assign a string to an existing variable, in the text box that displays the expression you configured, select the string you want to assign to an existing variable, right-click the selection, click **Use existing Variable**, and then click the name of the variable to which you want to assign the string.

  If a variable has multiple strings assigned to it, when you specify a new value for the variable during import, all strings assigned to the variable are replaced with the new value.

7. Click **Close**.

Modifying an entity template

September 21, 2020

You can modify only the parameters, bindings, and pages configured for a template. The name and location of the template specified when the template was created cannot be changed. The NetScaler appliance does not provide you with the option of modifying a load balancing virtual server template.

To modify an entity template by using the AppExpert feature node

1. Navigate to **AppExpert** > **Templates**.
2. In the details pane, on the **Entity Templates** tab, select the template you want to change, and then click **Open**.
3. In the **Modify…Template** dialog box, follow the instructions to modify a template.
4. Click **Finish**, and then click **Exit**.

To modify an entity template by using its corresponding feature node

1. Navigate to **Traffic Management**, select the feature (for example, Content Switching), and then select the entity (for example, Virtual Servers) for which you want to modify the entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the** Manage <`feature entity name`> Entity Templates** dialog box, select the template that you want to modify, and then click **Modify**.
4. In the **Modify <`template name`> Template** dialog box, follow the instructions to modify a template.
5. Click **Finish**, and then click **Exit**.
6. Click **Close**.

Deleting an entity template

September 21, 2020

Deleting an entity template does not affect any objects that have been created by using the template. You can delete a load balancing virtual server template only from the AppExpert feature node.

To delete an entity template by using the AppExpert feature node

1. Navigate to **AppExpert** > **Templates**.
2. In the details pane, on the **Entity Templates** tab, click the template you want to delete, and then click **Remove**.

To delete an entity template by using its corresponding feature node

1. Navigate to **Traffic Management**, and select the feature (for example, Content Switching) and then select the entity (for example, Virtual Servers), for which you want to delete the entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage…Entity Templates** dialog box, select the template that you want to delete, and then click **Delete**.

Creating an entity from a template

September 21, 2020

You can create an entity from an entity template either from the AppExpert feature node in the NetScaler GUI or from the NetScaler feature node that corresponds to the type of entity that you want to create. For example, you can create a content switching virtual server from a template with either the AppExpert feature node or the content switching feature node in the GUI.

The procedure for creating a load balancing virtual server from a template is different than the AppExpert procedure for creating other entities from templates.

After you create an instance of an entity using an entity template, you can configure it in the same way that you would any other object of that type, such as by using the GUI or the command line.

To create an entity from a template by using the AppExpert feature node

1. Navigate to **AppExpert** > **Templates**.
2. In the details pane, do one of the following:
   a) To create any entity other than a load balancing virtual server from a template, on the **Entity Templates** tab, click the template that you want to use, and then click Use Template.
   b) To create a load balancing virtual server from a template, on the LB Templates tab, click the template that you want to use, and then click **Use Template**.
3. In the <`Entity Template Name`> wizard, follow the instructions to create the entity on the NetScaler.
4. Click **Finish**, and then click **Exit**.

To create an entity from a template by using its corresponding feature node

1. Navigate to **Traffic Management**, and expand a feature node (for example, Content Switching), and then click an entity subnode (for example, Virtual Servers).

2. At the top of the details pane, click **Entity Templates**, and then click Use **Template**.

3. Click the name of the template that you want to use.

4. In the Use <`template name`> Template wizard, follow the instructions to create the entity.

   Only templates that match the current context are displayed. For example, in the details pane for content switching virtual servers, only entity templates for content switching virtual servers appear, if configured.

5. Click **Finish**, and then click **Exit**.

To create a load balancing virtual server by using a load balancing virtual server template

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.

2. In the details pane, click Use **Template**.
3. In the **Entity Template Wizard**, follow the instructions to create a load balancing virtual server on the NetScaler.

   Only templates that match the current context are displayed. For example, when you click Browse (Appliance), only entity templates for load balancing virtual servers appear, if configured.
4. Click **Finish**, and then click **Exit**.

   > **Note**: The **Entity Template Wizard** includes a **Specify Variable Values** page on which you can specify new values for variables. For more information about configuring variables in load balancing virtual server templates, see Configuring Variables in Load Balancing Virtual Server Templates."

Managing entity template folders

September 21, 2020

You can organize only load balancing virtual server template folders.

To organize load balancing virtual server template folders

1. Navigate to **AppExpert** > **Templates** > **LB Templates**.
2. In the **Manage LB Templates** dialog box do one of the following:
   - To change the name of a folder, select the folder and click **Rename**.

     You can also click the folder that you want to rename, and then press F2. You cannot rename the top-level default folder.
   - To remove the folder, select the folder and click **Delete**.

     You can also click the folder that you want to remove, and then press the Delete key. You cannot remove the top-level default folder.
3. Click **Close**.

Uploading and downloading entity templates

September 21, 2020

You can import the entity templates that are stored on your local computer. You can also download entity templates from the NetScaler appliance to your local computer and then import them to other NetScaler appliances.

To upload an entity template to the NetScaler appliance

1. Navigate to **Traffic Management**, and expand a feature node (for example, Content Switching), and then click a subnode (for example, Virtual Servers) for which you want to upload an entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage…Entity Templates** dialog box, click the top-level folder, and then click **Upload**.
4. In the **Upload Entity Template** dialog box, navigate to the template file that you want to upload, and then click **Select**.
5. Click **Close**.

To download an entity template from the NetScaler appliance

1. Navigate to **Traffic Management**, and expand a feature node (for example, Content Switching), and then click a subnode (for example, Virtual Servers) for which you want to upload an entity template.
2. At the top of the details pane, click **Entity Templates**, and then click **Manage Template**.
3. In the **Manage…Entity Templates** dialog box, click the template that you want to download, and then click **Download**.
4. In the **Download Entity Template** dialog box, navigate to the location at which you want to save the template on your local computer, enter a file name, and then click **Save**.
5. Click **Close**.

Understanding load balancing entity templates and deployment files

September 21, 2020

Load balancing entity templates are created in the same way that NetScaler application templates are created. When you export a load balancing virtual server to a template file, the following two files are automatically created:

- **Load balancing virtual server template file.** Contains XML elements that store the values of the parameters that are configured for the load balancing virtual server. The file also contains XML elements for storing information about bound policies.
- **Deployment file.** Contains XML elements that store deployment-specific information such as services, service groups, and configured variables.

In the template and deployment files, each unit of configuration information is encapsulated in a specific XML element that is meant for that unit type. For example, the load balancing method parameter, lbMethod, is encapsulated within the
`<lbmethod>` and
`<lbmethod>` tags.

Note: After you export a load balancing virtual server, you can add elements, remove elements, and modify existing elements before importing the configuration information to a NetScaler appliance.

Example of a load balancing virtual server template

Following is an example of a template file that was created from a load balancing virtual server called “Lbvip”:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <template>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <entitytemplate>
      <lbvserver_list>
        <lbvserver>
          <name>Lbvip</name>
          <servicetype>HTTP</servicetype>
          <ipv46>0.0.0.0</ipv46>
          <ipmask>*</ipmask>
          <port>0</port>
          <range>1</range>
          <persistencetype>NONE</persistencetype>
          <timeout>2</timeout>
          <persistencebackup>NONE</persistencebackup>
          <backuppersistencetimeout>2</backuppersistencetimeout>
          <lbmethod>LEASTCONNECTION</lbmethod>
          <persistmask>255.255.255.255</persistmask>
          <v6persistmasklen>128</v6persistmasklen>
          <pq>OFF</pq>
          <sc>OFF</sc>
          <m>IP</m>
          <datalength>0</datalength>
          <dataoffset>0</dataoffset>
          <sessionless>DISABLED</sessionless>
          <state>ENABLED</state>
          <connfailover>DISABLED</connfailover>
          <clttimeout>180</clttimeout>
          <somethod>NONE</somethod>
          <sopersistence>DISABLED</sopersistence>
          <sopersistencetimeout>2</sopersistencetimeout>
          <redirectportrewrite>DISABLED</redirectportrewrite>
          <downstateflush>DISABLED</downstateflush>
          <gt2gb>DISABLED</gt2gb>
          <ipmapping>0.0.0.0</ipmapping>
          <disableprimaryondown>DISABLED</disableprimaryondown>
          <insertvserveripport>OFF</insertvserveripport>
          <authentication>OFF</authentication>
          <authn401>OFF</authn401>
          <push>DISABLED</push>
          <pushlabel>none</pushlabel>
          <l2conn>OFF</l2conn>
          <appflowlog>DISABLED</appflowlog>
          <icmpvsrresponse>PASSIVE</icmpvsrresponse>
          <lbvserver_cmppolicy_binding_list>
            <lbvserver_cmppolicy_binding>
              <name>Lbvip</name>
              <policyname>NOPOLICY-COMPRESSION</policyname>
              <priority>100</priority>
              <gotopriorityexpression>END</gotopriorityexpression>
              <bindpoint>REQUEST</bindpoint>
            </lbvserver_cmppolicy_binding>
          </lbvserver_cmppolicy_binding_list>
        </lbvserver>
      </lbvserver_list>
    </entitytemplate>
  </template>
<!--NeedCopy-->
```

Example of a deployment file

Following is the deployment file associated with the virtual server in the preceding example:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <template_deployment>
    <template_info>
      <entity_name>Lbvip</entity_name>
      <version_major>10</version_major>
      <version_minor>0</version_minor>
      <build_number>40.406</build_number>
    </template_info>
    <service_list>
      <service>
      <ip>1.2.3.4</ip>
      <port>80</port>
      <servicetype>HTTP</servicetype>
      </service>
    </service_list>
    <servicegroup_list>
      <servicegroup>
        <name>svcgrp</name>
        <servicetype>HTTP</servicetype>
        <servicegroup_servicegroupmember_binding_list>
          <servicegroup_servicegroupmember_binding>
            <ip>1.2.3.90</ip>
            <port>80</port>
          </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.8.0</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.8.1</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
        <servicegroup_servicegroupmember_binding>
          <ip>1.2.9.0</ip>
          <port>80</port>
        </servicegroup_servicegroupmember_binding>
      </servicegroup_servicegroupmember_binding_list>
    </servicegroup>
  </servicegroup_list>
</template_deployment>
<!--NeedCopy-->
```

HTTP callouts

September 21, 2020

For certain types of requests, or when certain criteria are met during policy evaluation, you might want to stall policy evaluation briefly, retrieve information from a server, and then perform a specific action that depends on the information that is retrieved. At other times, when you receive certain types of requests, you might want to update a database or the content hosted on a Web server. HTTP callouts enable you to perform all these tasks.

An HTTP callout is an HTTP or HTTPS request that the NetScaler appliance generates and sends to an external application when certain criteria are met during policy evaluation. The information that is retrieved from the server can be analyzed by default syntax policy expressions, and an appropriate action can be performed. You can configure HTTP callouts for HTTP content switching, TCP content switching, rewrite, responder, and for the token-based method of load balancing.

Before you configure an HTTP callout, you must set up an application on the server to which the callout will be sent. The application, which is called the *HTTP callout agent* , must be configured to respond to the HTTP callout request with the required information. The HTTP callout agent can also be a Web server that serves the data for which the NetScaler appliance sends the callout. You must make sure that the format of the response to an HTTP callout does not change from one invocation to another.

After you set up the HTTP callout agent, you configure the HTTP callout on the NetScaler appliance. Finally, to invoke the callout, you include the callout in a default syntax policy in the appropriate NetScaler feature and then bind the policy to the bind point at which you want the policy to be evaluated.

After you have configured the HTTP callout, you must verify the configuration to make sure that the callout is working correctly.

How an HTTP callout works

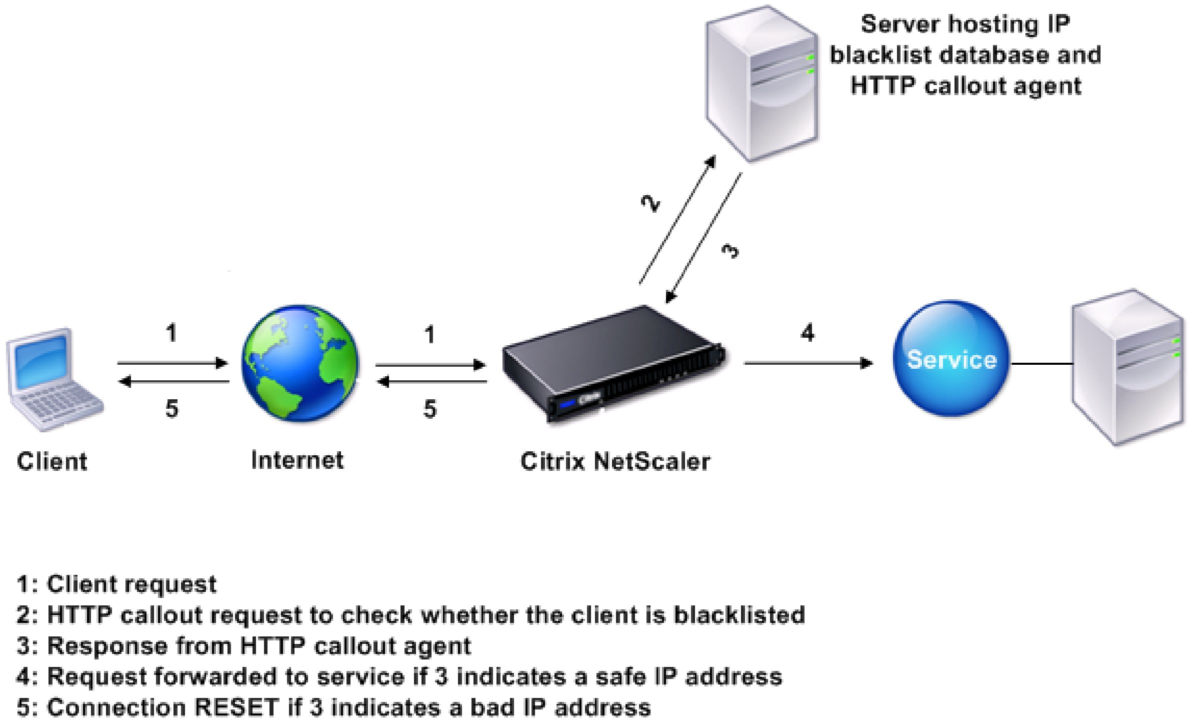
September 21, 2020

When the NetScaler appliance receives a client request, the appliance evaluates the request against the policies bound to various bind points. During this evaluation, if the appliance encounters the HTTP callout expression, SYS.HTTP_CALLOUT(<name>), it stalls policy evaluation briefly and sends a request to the HTTP callout agent by using the parameters configured for the specified HTTP callout. Upon receiving the response, the appliance inspects the specified portion of the response, and then either performs an action or evaluates the next policy, depending on whether the evaluation of the response from the HTTP callout agent evaluates to TRUE or FALSE, respectively. For example, if the HTTP callout is included in a responder policy, if the evaluation of the response evaluates to TRUE, the appliance performs the action associated with the responder policy.

If the HTTP callout configuration is incorrect or incomplete, or if the callout invokes itself recursively, the appliance raises an UNDEF condition, and updates the undefined hits counter.

The following figure illustrates the working of an HTTP callout that is invoked from a globally bound responder policy. The HTTP callout is configured to include the IP address of the client that is associated with an incoming request. When the NetScaler appliance receives a request from a client, the appliance generates the callout request and sends it to the callout server, which hosts a database of blacklisted IP addresses and an HTTP callout agent that checks whether the client's IP address is listed in the database. The HTTP callout agent receives the callout request, checks whether the client's IP address is listed, and sends a response that the NetScaler appliance evaluates. If the response indicates that the client's IP address is not blacklisted, the appliance forwards the response to the configured service. If the client's IP address is blacklisted, the appliance resets the client connection

Figure 1. HTTP Callout Entity Model


Server hosting IP blacklist database and HTTP callout agent
2
3
1
5
1
5
4
Service
Client
Internet
Citrix NetScaler
1: Client request
2: HTTP callout request to check whether the client is blacklisted
3: Response from HTTP callout agent
4: Request forwarded to service if 3 indicates a safe IP address
5: Connection RESET if 3 indicates a bad IP address

Notes on the format of HTTP requests and responses

September 21, 2020

The NetScaler appliance does not check for the validity of the HTTP callout request. Therefore, before you configure HTTP callouts, you must know the format of an HTTP request. You must also know the format of an HTTP response, because configuring an HTTP callout involves configuring expressions that evaluate the response from the HTTP callout agent.

This section includes the following sections:

- Format of an HTTP Request

- Format of an HTTP Response

Format of an HTTP request

An HTTP request contains a series of lines that each end with a carriage return and a line feed, represented as either `<CR><LF>` or `\r\n`.

The first line of a request (the *message line*) contains the HTTP method and target. For example, a message line for a GET request contains the keyword GET and a string that represents the object that is to be fetched, as shown in the following example:

```
GET /mysite/mydirectory/index.html HTTP/1.1\r\n
<!--NeedCopy-->
```

The rest of the request contains HTTP headers, including a required Host header and, if applicable, a message body.

The request ends with a bank line (an extra <CR><LF> or \r\n).

Following is an example of a request:

```
Get /mysite/index.html HTTP/1.1\r\n
Host: 10.101.101.10\r\n
Accept: */*\r\n
\r\n
<!--NeedCopy-->
```

Format of an HTTP Response

An HTTP response contains a status message, response HTTP headers, and the requested object or, if the requested object cannot be served, an error message.

Following is an example of a response:

```
HTTP/1.1 200 OK\r\n
Content-Length: 55\r\n
Content-Type: text/html\r\n
Last-Modified: Wed, 12 Aug 1998 15:03:50 GMT\r\n
Accept-Ranges: bytes\r\n
ETag: “04f97692cbd1:377” \r\n
Date: Thu, 19 Jun 2008 19:29:07 GMT\r\n
\r\n
<55-character response>
<!--NeedCopy-->
```

Configuring an HTTP callout

September 21, 2020

When configuring an HTTP callout, you specify the type of request (HTTP or HTTPS), destination and format of the request, the expected format of the response, and, finally, the portion of the response that you want to analyze.

For the destination, you either specify the IP address and port of the HTTP callout agent or engage a load balancing, content switching, or cache redirection virtual server to manage the HTTP callout requests. In the first case, the HTTP callout requests will be sent directly to the HTTP callout agent. In the second case, the HTTP callout requests will be sent to the virtual IP address (VIP) of the specified virtual server. The virtual server will then process the request in the same way as it processes a client request. For example, if you expect a large number of callouts to be generated, you can configure instances of the HTTP callout agent on multiple servers, bind these instances (as services) to a load balancing virtual server, and then specify the load balancing virtual server in the HTTP callout configuration. The load balancing virtual server then balances the load on those configured instances as determined by the load balancing algorithm.

For the format of the HTTP callout request, you can specify the individual attributes of the HTTP callout request (an attribute-based HTTP callout), or you can specify the entire HTTP callout request as a default syntax expression (an expression-based HTTP callout).

> **Note**:
>
> The appliance does not check for the validity of the request. You must make sure that the request is a valid request. An incorrect or incomplete HTTP callout configuration results in a runtime UNDEF condition that is not associated with an action. The UNDEF condition merely updates the Undefined Hits counter, which enables you to troubleshoot an incorrectly configured HTTP callout. However, the appliance parses the HTTP callout request to enable you to configure certain NetScaler features for the callout. This can lead to an HTTP callout invoking itself. For information about callout recursion and how you can avoid it, see “
> Avoiding HTTP Callout Recursion.”

Finally, regardless of whether you use HTTP request attributes or an expression to define the format of the HTTP callout request, you must specify the format of the response from the HTTP callout agent and the portion of the response that you want to evaluate. The response can be a Boolean value, a number, or text. The portion of the response that you want to evaluate is specified by an expression. For example, if you specify that the response contains text, you can use HTTP.RES.BODY(<unit>) to specify that the appliance must evaluate only the first <unit> bytes of the response from the callout agent.

At the command line, you first create an HTTP callout by using the

add command. When you add a callout, all parameters are set to a default value of NONE, except the HTTP method, which is set to a default value of GET. You then configure the callout's parameters by using the set command. The
set command is used to configure both types of callouts (attribute-based and expression-based). The difference lies in the parameters that are used for configuring the two types of callouts. Accordingly, the command-line instructions that follow include a
set command for configuring an attribute-based callout and a
set command for configuring an expression-based callout. In the GUI, all of these configuration tasks are performed in a single dialog box.

Note: Before you put an HTTP callout into a policy, you can modify all configured parameters except the return type. Once an HTTP callout is in a policy, you cannot completely modify an expression that is configured in the callout. For example, you cannot change

```
HTTP.REQ.HEADER("myval") to
CLIENT.IP.SRC. However, you can modify the operators and arguments that
    are passed to the expression. For example, you can change
HTTP.REQ.HEADER("myVal1") to
HTTP.REQ.HEADER("myVal2"), or
HTTP.REQ.HEADER("myVal") to
HTTP.REQ.HEADER("myVal").AFTER_STR(<string>). If the set command fails,
    create a new HTTP callout.
<!--NeedCopy-->
```

HTTP callout configuration involves configuring default syntax expressions.

To configure an HTTP callout by using the command line interface

At the command prompt, do the following:

1. Create a HTTP callout.

`add policy httpCallout <name>`

```
**Example**
```

```
    add policy httpCallout mycallout
<!--NeedCopy-->
```

1. Configure the details of the HTTP callout.

   - To configure an attribute-based HTTP callout, type:

```
set policy httpCallout <name> [-IPAddress <ip_addr|ipv6_addr|*>] [-port
    <port|*>] [-vServer <string>] [-returnType <returnType>] [-
    httpMethod ( GET | POST )] [-hostExpr <string>] [-urlStemExpr <
    string>] [-headers <name(value)> ...] [-parameters <name(value)>
    ...] [-resultExpr <string>]
<!--NeedCopy-->
```

Example:

```
> set policy httpCallout mycallout -vserver lbv1 -returnType num -
    httpMethod GET -hostExpr 'http.req.header("Host")'
-urlStemExpr "http.req.url" -parameters Name("My Name") -headers Name("
    MyHeader")
-resultExpr "http.res.body(10000).length"

<!--NeedCopy-->
```

To configure an expression-based HTTP callout, type:

```
set policy httpCallout <name> [-vServer <string>] [-returnType <
    returnType>] [-httpMethod ( GET | POST )] [-fullReqExpr <string>] [-
    resultExpr <string>]
<!--NeedCopy-->
```

Example:

```
> set policy httpCallout mycallout1 -vserver lbv1 -returnType num -
    httpMethod GET
-fullReqExpr q{
 "GET " + http.req.url + "HTTP/" + http.req.version.major + "." + http.
     req.version.minor.sub(1)+
"r\nHost:10.101.10.10\r\nAccept: */*\r\n\r\n" }

<!--NeedCopy-->
```

3. Verify the configurations of the HTTP callout.

```
show policy httpCallout <name>
<!--NeedCopy-->
```

To configure an HTTP callout by using the GUI

1. Navigate to **AppExpert** > **HTTP Callouts**.
2. In the details pane, click **Add**.
3. In the **Create HTTP Callout** dialog box, configure the parameters of the HTTP callout. For a description of the parameter, hover the mouse cursor over the check box.

4. Click **Create** and then click **Close**.

Verifying the Configuration

September 21, 2020

For an HTTP callout to work correctly, all the HTTP callout parameters and the entities associated with the callout must be configured correctly. While the NetScaler appliance does not check the validity of the HTTP callout parameters, it indicates the state of the bound entities, namely the server or virtual server to which the HTTP callout is sent. The following table lists the icons and describes the conditions under which the icons are displayed.

Icon	Indicates that
🟢	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is UP.
🟡	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is OUT OF SERVICE.
🔴	The state of the server that hosts the HTTP callout agent, or the load balancing, content switching, or cache redirection virtual server to which the HTTP callout is sent is DOWN.

For an HTTP callout to function correctly, the icon must be green at all times. If the icon is not green, check the state of the callout server or virtual server to which the HTTP callout is sent. If the HTTP callout is not working as expected even though the icon is green, check the parameters configured for the callout.

You can also verify the configuration by sending test requests that match the policy from which the HTTP callout is invoked, checking the hits counter for the policy and the HTTP callout, and verifying the responses that the NetScaler appliance sends to the client.

> **Note**:
>
> An HTTP callout can sometimes invoke itself recursively a second time. If this happens, the hits counter is incremented by two counts for each callout that is generated by the appliance. For the hits counter to display the correct value, you must configure the HTTP callout in such a way that it does not invoke itself a second time. For more information about how you can avoid HTTP callout recursion, see
> Avoiding HTTP Callout Recursion.

To view the hits counter for an HTTP callout

1. Navigate to AppExpert > HTTP Callouts.
2. In the details pane, click the HTTP callout for which you want to view the hits counter, and then view the hits in the Details area.

Invoking an HTTP Callout

September 21, 2020

After you configure an HTTP callout, you invoke the callout by including the `SYS.HTTP_CALLOUT(<name>)` expression in a default syntax policy rule. In this expression, `<name>` is the name of the HTTP callout that you want to invoke.

You can use default syntax expression operators with the callout expression to process the response and then perform an appropriate action. The return type of the response from the HTTP callout agent determines the set of operators that you can use on the response. If the part of the response that you want to analyze is text, you can use a text operator to analyze the response. For example, you can use the CONTAINS(<string>) operator to check whether the specified portion of the response contains a particular string, as in the following example:

```
SYS.HTTP_CALLOUT(mycallout).contains("Good IP address")
<!--NeedCopy-->
```

If you use the preceding expression in a responder policy, you can configure an appropriate responder action.

Similarly, if the part of the response that you want to evaluate is a number, you can use a numeric operator such as GT(int). If the response contains a Boolean value, you can use a Boolean operator.

Note: An HTTP callout can invoke itself recursively. HTTP callout recursion can be avoided by combining the HTTP callout expression with a default syntax expression that prevents recursion. For in-

formation about how you can avoid HTTP callout recursion, see
Avoiding HTTP Callout Recursion.

You can also cascade HTTP callouts by configuring policies that each invoke a callout after evaluating previously generated callouts. In this scenario, after one policy invokes a callout, when the NetScaler appliance is parsing the callout before sending the callout to the callout server, a second set of policies can evaluate the callout and invoke additional callouts, which can in turn be evaluated by a third set of policies, and so on. Such an implementation is described in the following example.

First, you could configure an HTTP callout called myCallout1, and then configure a responder policy, Pol1, to invoke myCallout1. Then, you could configure a second HTTP callout, myCallout2, and a responder policy, Pol2. You configure Pol2 to evaluate myCallout1 and invoke myCallout2. You bind both responder policies globally.

To avoid HTTP callout recursion, myCallout1 is configured with a unique custom HTTP header called "Request1." Pol1 is configured to avoid HTTP callout recursion by using the default syntax expression,

```
HTTP.REQ.HEADER("Request1").EQ("Callout Request").NOT.
<!--NeedCopy-->
```

Pol2 uses the same default syntax expression, but excludes the .NOT operator so that the policy evaluates myCallout1 when the NetScaler appliance is parsing it. Note that myCallout2 identifies its own unique header called "Request2," and Pol2 includes a default syntax expression to prevent myCallout2 from invoking itself recursively.

Example:

```
> add policy httpCallout myCallout1

Done

> set policy httpCallout myCallout1 -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr
 ""10.102.3.95"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"" -headers Request1
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"

Done

> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\").NOT &&
SYS.HTTP_CALLOUT(myCallout1).CONTAINS(\"IP Matched\")" RESET

Done

```

```
> bind responder global Pol1 100 END -type OVERRIDE

Done

> add policy httpCallout myCallout2

Done

> set policy httpCallout myCallout2 -IPAddress 10.102.3.96 -port 80 -returnType TEXT -hostExpr
"\"10.102.3.96\"" -urlStemExpr "\"/cgi-bin/check_clnt_location_from_database.pl\"" -headers Request2
("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(200)"

 Done

> add responder policy Pol2 "HTTP.REQ.HEADER(\"Request2\").EQ(\"Callout Request\").NOT &&
 HTTP.REQ.HEADER(\"Request1\").EQ(\"Callout Request\") && SYS.HTTP_CALLOUT(myCallout2).CONTAINS
(\"APAC\")" RESET

 Done

> bind responder global Pol2 110 END -type OVERRIDE

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

Avoiding HTTP callout recursion

September 21, 2020

Even though the NetScaler appliance does not check for the validity of the HTTP callout request, it parses the request once before it sends the request to the HTTP callout agent. This parsing allows the appliance to treat the callout request as any other incoming request, which in turn allows you to configure several useful NetScaler features (such as integrated caching, SureConnect, and Priority Queuing) to work on the callout request.

However, during this parsing, the HTTP callout request can select the same policy and therefore invoke itself recursively. The appliance detects the recursive invocation and raises an undefined (UN-

DEF) condition. However, the recursive invocation results in the policy and HTTP callout select counters being incremented by two counts each instead of one count each.

To prevent a callout from invoking itself, you must identify at least one unique characteristic of the HTTP callout request, and then exclude all requests with this characteristic from being processed by the policy rule that invokes the callout. You can do so by including another default syntax expression in the policy rule. The expression must precede the `SYS.HTTP\\_CALLOUT(\<name\>)` expression so that it is evaluated before the callout expression is evaluated. For example:

```
<Expression that prevents callout recursion> OR SYS.HTTP_CALLOUT(<name>)
<!--NeedCopy-->
```

When you configure a policy rule in this way, when the appliance generates the request and parses it, the compound rule evaluates to FALSE, the callout is not generated a second time, and the select counters are incremented correctly.

One way by which you can assign a unique characteristic to an HTTP callout request is to include a unique custom HTTP header when you configure the callout. Following is an example of an HTTP callout called "myCallout." The callout generates an HTTP request that checks whether a client's IP address is present in a database of blacklisted IP addresses. The callout includes a custom header called "Request," which is set to the value "Callout Request." A globally bound responder policy, "Pol1," invokes the HTTP callout but excludes all requests whose Request header is set to this value, thus preventing a second invocation of myCallout. The expression that prevents a second invocation is HTTP.REQ.HEADER(\“Request\”).EQ(\“Callout Request\”).NOT.

Example:

```
> add policy httpCallout myCallout
 Done

> set policy httpCallout myCallout -IPAddress 10.102.3.95 -port 80 -returnType TEXT -hostExpr "\"10.102.3.95\"" -urlStemExpr "\"/cgi-bin/check_clnt_from_database.pl\"" -headers Request("Callout Request") -parameters cip(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(100)"
Done

> add responder policy Pol1 "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(myCallout).CONTAINS(\"IP Matched\")" RESET
Done

> bind responder global Pol1 100 END -type OVERRIDE
Done
```

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

> **Note**:
> You can also configure an expression to check whether the request URL includes the stem expression configured for the HTTP callout. To implement the solution, ensure the HTTP callout agent can respond only to HTTP callouts and not to other requests directed through the appliance. If the HTTP callout agent is an application or Web server that serves other client requests, such an expression prevents the appliance from processing those client requests. Instead, use a unique custom header as described earlier.

Caching HTTP Callout Responses

September 21, 2020

For improved performance while using callouts, you can use the integrated caching feature to cache callout responses. The responses are stored in an integrated caching content group named callout-ContentGroup for a specified time duration.

Note: To cache callout responses, make sure that the integrated caching feature is enabled.

To set the cache duration by using the command line interface

At the command prompt, type:

`set policy httpCallout <name> -cacheForSecs <secs>`

Example:

```
> set httpcallout httpcallout1 -cacheForSecs 120
<!--NeedCopy-->
```

To set the cache duration by using the GUI

1. Navigate to **AppExpert > **HTTP Callouts**.
2. In the details pane, select the HTTP callout for which you want to set the cache duration and click **Open**.
3. In the **Configure HTTP Callout** dialog box, specify the **Cache Expiration Time**.
4. Verify that you have entered the correct time duration, and then click **OK**.

Use Case: Filtering Clients by Using an IP Blacklist

September 21, 2020

HTTP callouts can be used to block requests from clients that are blacklisted by the administrator. The list of clients can be a publicly known blacklist, a blacklist that you maintain for you organization, or a combination of both.

The NetScaler appliance checks the IP address of the client against the pre-configured blacklist and blocks the transaction if the IP address has been blacklisted. If the IP address is not in the list, the appliance processes the transaction.

To implement this configuration, you must perform the following tasks:

1. Enable responder on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response to the HTTP callout, and then bind the policy globally.
4. Create an HTTP callout agent on the remote server.

Enabling Responder

You must enable responder before you can use it.

To enable responder by using the GUI

1. Make sure that you have installed the responder license.
2. In the GUI, expand AppExpert, and right-click Responder, and then click Enable Responder feature.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTP_Callout, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see
Configuring an HTTP Callout pdf.

Configuring a Responder Policy and Binding it Globally

After you configure the HTTP callout, verify the callout configuration, and then configure a responder policy to invoke the callout. While you can create a responder policy in the
Policies sub-node and then bind it globally by using the

Responder Policy Manager, this demonstration uses the
Responder Policy Manager to create the responder policy and bind the policy globally.

To create a responder policy and bind it globally by using the GUI

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, under **Policy Name**, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
    a) In Name, type **Policy-Responder-1**.
    b) In **Action**, select **RESET**.
    c) In **Undefined-Result Action**, select **Global undefined-result** action.
    d) In **Expression**, type the following default syntax expression:

    ```
    "HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT &&
        SYS.HTTP_CALLOUT(HTTP_Callout).CONTAINS(\"IP Matched\")"
    <!--NeedCopy-->
    ```

    e) Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You must now create an HTTP callout agent on the remote callout server that will receive callout requests from the NetScaler appliance and respond appropriately. The HTTP callout agent is a script that is different for each deployment and must be written with the server specifications in mind, such as the type of database and the scripting language supported.

Following is a sample callout agent that verifies whether the given IP address is part of an IP blacklist. The agent has been written in the Perl scripting language and uses a MYSQL database.

The following CGI script checks for a given IP address on the callout server.

```
#!/usr/bin/perl -w
print "Content-type: text/html\n\n";
        use DBI();
        use CGI qw(:standard);
#Take the Client IP address from the request query
        my $ip_to_check = param('cip');
# Where a MYSQL database is running
        my $dsn = 'DBI:mysql:BAD_CLIENT:localhost';
# Database username to connect with
        my $db_user_name =  ‘dbuser’ ;
```

```
# Database password to connect with
        my $db_password = 'dbpassword';
        my ($id, $password);
# Connecting to the database
        my $dbh = DBI->connect($dsn, $db_user_name, $db_password);
        my $sth = $dbh->prepare(qq{
  select * from bad_clnt  }
 );
        $sth->execute();
        while (my ($ip_in_database) = $sth->fetchrow_array()) {

        chomp($ip_in_database);
# Check for IP match
        if ($ip_in_database eq $ip_to_check) {

              print "\n IP Matched\n";
                                                     $sth->finish();
                                               exit;
                 }

        }

       print "\n IP Failed\n";
       $sth->finish();
       exit;
<!--NeedCopy-->
```

Use Case: ESI Support for Fetching and Updating Content Dynamically

September 21, 2020

Edge Side Includes (ESI) is a markup language for edge-level dynamic Web content assembly. It helps in accelerating dynamic Web-based applications by defining a simple markup language to describe cacheable and non-cacheable Web page components that can be aggregated, assembled, and delivered at the network edge. By using HTTP callouts on the NetScaler appliance, you can read through the ESI constructs and aggregate or assemble content dynamically.

To implement this configuration, you must perform the following tasks:

1. Enable rewrite on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.

3. Configure a rewrite action to replace the ESI content with the callout response body.
4. Configure a rewrite policy to specify the conditions under which the action is performed, and then bind the rewrite policy globally.

Enabling Rewrite

Rewrite must be enabled before it is used on the NetScaler appliance. The following procedure describes the steps to enable the rewrite feature.

To enable rewrite by using the GUI

1. Make sure that you have installed the rewrite license.
2. In the GUI, expand AppExpert, and right-click Rewrite, and then click Enable Rewrite feature.

Creating an HTTP Callout on the NetScaler Appliance

For more information about creating an HTTP callout, see
Configuring an HTTP Callout.
For more information about the parameter values, see Parameters and Values for HTTP-Callout-2 pdf.

Configuring the rewrite action

Create a rewrite action, Action-Rewrite-1, to replace the ESI content with the callout response body. Use the parameter settings shown in the following table.

Table 2. Parameters and Values for Action-Rewrite-1

Parameter	Value
Name	Action-Rewrite-1
Type	Replace
Expression to choose target text reference	"HTTP.RES.BODY(500).AFTER_STR (\"<example>\").BEFORE_STR (\"</example>\")"
String expression for replacement text	"SYS.HTTP_CALLOUT(HTTP-Callout-2)"

To configure the rewrite action by using the GUI

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.

2. In the details pane, click **Add**.

3. In the **Create Rewrite Action** dialog box, in Name, type **Action-Rewrite-1**.
4. In Type, select **REPLACE**.
5. In **Expression** to choose target text reference, type the following default syntax expression:

```
"HTTP.RES.BODY(500).AFTER_STR(\"<example>\").BEFORE_STR(\"</
    example>\")"
<!--NeedCopy-->
```

6. In the String expression for replacement text, type the following string expression:

```
"SYS.HTTP_CALLOUT(HTTP-Callout-2)"
<!--NeedCopy-->
```

7. Click **Create**, and then click **Close**.

Creating the rewrite policy and binding it globally

Create a rewrite policy, Policy-Rewrite-1, with the parameter settings shown in the following table. You can create a rewrite policy in the
Policies subnode and then bind it globally by using the
Rewrite Policy Manager. Alternatively, you can use the
Rewrite Policy Manager to perform both these tasks simultaneously. This demonstration uses the Rewrite Policy Manager to perform both tasks.

Parameter	Value
Name	Policy-Rewrite-1
Action	Action_Rewrite-1
Undefined Result Action	-Global undefined-result action-
Expression	“HTTP.REQ.HEADER(”Name”).CONTAINS (”Callout”).NOT”

Table 3. Parameters and Values for Policy-Rewrite-1

To configure a rewrite policy and bind it globally by using the GUI

1. Navigate to **AppExpert** > **Rewrite**.
2. In the details pane, under **Policy Manager**, click **Rewrite Policy Manager**.
3. In the **Rewrite Policy Manager** dialog box, click **Override Global**.

4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Rewrite Policy** dialog box, do the following:

   1.In Name, type Policy-Rewrite-1.

   a) In Action, select Action-Rewrite-1.
   b) In Undefined-Result Action, select Global undefined-result action.
   c) In Expression, type the following default syntax expression:

```
"HTTP.REQ.HEADER(\"Name\").CONTAINS(\"Callout\").NOT"
<!--NeedCopy-->
```

5. Click **Create**, and then click **Close**.

6. Click **Apply Changes**, and then click **Close**.

Use Case: Access control and authentication

September 21, 2020

In high security zones, it is mandatory to externally authenticate the user before a resource is accessed by clients. On the NetScaler appliance, you can use HTTP callouts to externally authenticate the user by evaluating the credentials supplied. In this example, the assumption is that the client is sending the user name and password through HTTP headers in the request. However, the same information could be fetched from the URL or the HTTP body.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the appliance and configure it with details about the external server and other required parameters.
3. Configure a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling responder

The responder feature must be enabled before it is used on the NetScaler appliance.

To enable responder by using the GUI

1. Make sure that the responder license is installed.
2. In the GUI, expand **AppExpert**, and right-click **Responder**, and then click **Enable Responder** feature.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTPCallout, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see Configuring an HTTP Callout.

Refer to HTTP callout table, for more information about its parameters and values.

Creating a Responder policy to analyze the response

Create a responder policy, Policy-Responder-3, that will check the response from the callout server and RESET the connection if the source IP address has been blacklisted. Create the policy with the parameters settings shown in the following table. While you can create a responder policy in the Policies subnode and then bind it globally by using the
Responder Policy Manager, this demonstration uses the
Responder Policy Manager to create the responder policy and bind the policy globally.

Parameter	Value
Name	Policy-Responder-3
Action	RESET
Undefined-Result-Action	-Global undefined-result action-
Expression	“HTTP.REQ.HEADER(\“Request\”).EQ(\“Callout Request\”).NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\“Authentication Failed\”)”

Table 2. Parameters and Values for Policy-Responder-3

To create a responder policy and bind it globally by using the GUI

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder Policy Manager.**
3. In the **Responder Policy Manger** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the Policy Name column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
    a) In Name, type Policy-Responder-3.
    b) In **Action**, select **RESET**.
    c) In Undefined-Result Action , select Global undefined-result action.
    d) In the **Expression** text box, type:

```
"HTTP.REQ.HEADER(\"Request\").EQ(\"Callout Request\").NOT && SYS.HTTP_CALLOUT(HTTP-Callout-3).CONTAINS(\"Authentication Failed\")"
<!--NeedCopy-->
```

    e) Click **Create**, and then click **Close**.

6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout agent on the remote server

You now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds appropriately. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

Following is sample callout agent pseudo-code that verifies whether the supplied user name and password are valid. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To verify the supplied user name and password by using pseudo-code

1. Accept the user name and password supplied in the request and format them appropriately.
2. Connect to the database that contains all the valid user names and passwords.
3. Check the supplied credentials against your database.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

Use Case: OWA-Based Spam Filtering

September 21, 2020

Spam filtering is the ability to dynamically block emails that are not from a known or trusted source or that have inappropriate content. Spam filtering requires an associated business logic that indicates that a particular kind of message is spam. When the NetScaler appliance processes Outlook Web Access (OWA) messages based on the HTTP protocol, HTTP callouts can be used to filter spam.

You can use HTTP callouts to extract any portion of the incoming message and check with an external callout server that has been configured with rules that are meant for determining whether a message is legitimate or spam. In case of spam email, for security reasons, the NetScaler appliance does not notify the sender that the email is marked as spam.

The following example conducts a very basic check for various listed keywords in the email subject. These checks can be more complex in a production environment.

To implement this configuration, you must perform the following tasks:

1. Enable the responder feature on the NetScaler appliance.
2. Create an HTTP callout on the NetScaler appliance and configure it with details about the external server and other required parameters.
3. Create a responder policy to analyze the response, and then bind the policy globally.
4. Create a callout agent on the remote server.

Enabling Responder

The responder feature must be enabled before it can be used on the NetScaler appliance.

To enable responder by using the GUI

1. Make sure that the responder license is installed.
2. In the GUI, expand AppExpert, and right-click Responder, and then click **Enable Responder** feature.

Creating an HTTP Callout on the NetScaler Appliance

Create an HTTP callout, HTTPCallout, with the parameter settings shown in the following table. For more information about creating an HTTP callout, see
Configuring an HTTP Callout.

For more information, see Parameters and Values for HTTCallout pdf.

Creating a Responder Action

Create a responder action, Action-Responder-4. Create the action with the parameter settings shown in the following table.

Parameter	Value
Name	Action-Responder-4
Type	Respond with

Parameter	Value
Target	””HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n””

Table 2. Parameters and Values for Action-Responder-4

To create a responder action by using the GUI

1. Navigate to **AppExpert** > **Responder** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, in Name, type **Action-Responder-4**.
4. In Type, click **Respond with**.
5. In Target, type:

```
    "\"HTTP/1.1 200 OK\r\nServer: Microsoft-IIS/6.0\r\nX-Powered-By: ASP.NET\r\nContent-Length: 0\r\nMS-WebStorage: 6.5.6944\r\nCache-Control: no-cache\r\n\r\n\""
<!--NeedCopy-->
```

1. Click **Create**, and then click **Close**.

Creating a Responder Policy to Invoke the HTTP Callout

Create a responder policy, Policy-Responder-4, that will check the request body and, if the body contains the word “ *subject*,” invoke the HTTP callout to verify the email. Create the policy with the parameter settings shown in the following table. While you can create a responder policy in the Policies subnode and then bind it globally by using the Responder Policy Manager, this demonstration uses the Responder Policy Manager to create the responder policy and bind it globally.

Parameter	Value
Name	Policy-Responder-4
Action	Action-Responder-4

Parameter	Value
Undefined-Result-Action	-Global undefined-result action-
Expression	“HTTP.REQ.BODY(1000).CONTAINS(”urn:schemas:httpmail:su && SYS.HTTP_CALLOUT(HTTP-Callout-4)”

To create a responder policy by using the GUI

1. Navigate to **AppExpert** > **Responder**.
2. In the details pane, under **Policy Manager**, click **Responder policy** manager.
3. In the **Responder Policy Manger** dialog box, click **Override Global**.
4. Click **Insert Policy**, and then, in the **Policy Name** column, click **New Policy**.
5. In the **Create Responder Policy** dialog box, do the following:
    a) In Name, type **Policy-Responder-4**.
    b) In Action, click **Action-Responder-4**.
    c) In Undefined-Result Action, click **Global undefined-result** action.
    d) In the **Expression** text box, type:

    ```
    "HTTP.REQ.BODY(1000).CONTAINS(\"urn:schemas:httpmail:subject\") && SYS.HTTP_CALLOUT(HTTP-Callout-4)"
    <!--NeedCopy-->
    ```

    e) Click **Create**, and then click **Close**.
6. Click **Apply Changes**, and then click **Close**.

Creating an HTTP Callout Agent on the Remote Server

You will now need to create an HTTP callout agent on the remote callout server. The HTTP callout agent receives callout requests from the NetScaler appliance and responds accordingly. The callout agent is a script that is different for each deployment and must be written with server specifications in mind, such as the type of database and the scripting language supported.

The following pseudo-code provides instructions for creating a callout agent that checks a list of words that are generally understood to indicate spam mails. The agent can be implemented in any programming language of your choice. The pseudo-code is to be used only as a guideline for developing the callout agent. You can build additional functionality into the program.

To identify spam email by using pseudo-code

1. Accept the email subject provided by the NetScaler appliance.
2. Connect to the database that contains all the terms against which the email subject is checked.

3. Check the words in the email subject against the spam word list.
4. Format the response as required by the HTTP callout.
5. Send the response to the NetScaler appliance.

Use Case: Dynamic Content Switching

September 21, 2020

This use case provides dynamic content switching by using an HTTP callout to get the name of the load balancing virtual server to which the request is forwarded.

1. Add a content switching virtual server.

   ```
   > add cs vserver cs_vserver1 HTTP 10.102.29.196 80
   <!--NeedCopy-->
   ```

2. Create an HTTP callout.

   ```
   > add policy httpCallout http_callout1
   <!--NeedCopy-->
   ```

3. Configure the HTTP callout to respond with the name of the load balancing virtual server from a request that contains the client IP address in the HTTP header “X-CLIENT-IP”.

   ```
   > set policy httpCallout http_callout1 -IPAddress 10.217.14.23 -port 80 -returnType TEXT -hostExpr "\"www.get-lbvip.com\"" -urlStemExpr "\"/index.html\"" -headers X-CLIENT-IP(CLIENT.IP.SRC) -resultExpr "HTTP.RES.BODY(1000).AFTER_STR(\"<lbvip>\").BEFORE_STR(\"</lbvip\")"
   <!--NeedCopy-->
   ```

4. Configure the content switching action to retrieve the callout response.

   ```
   > add cs action cs_action1 -targetVserverExpr 'SYS.HTTP_CALLOUT(http_callout1)'
   <!--NeedCopy-->
   ```

   Note: You must bind a load balancing virtual server to the content switching virtual server to account for:

   - The non-availability of the load balancing virtual server that the callout resolves to.
   - A UNDEF condition that results from the execution of the callout.

```
> bind cs vserver cs_vserver1 -lbvserver default_lbvip
<!--NeedCopy-->
```

5. Configure the content switching policy.

```
> add cs policy cs_policy1 -rule true -action cs_action1
<!--NeedCopy-->
```

6. Binding the content switching policy to the content switching virtual server.

```
> bind cs vserver cs_vserver1 -policyName cs_policy1 -priority 10
<!--NeedCopy-->
```

Pattern sets and data sets

September 21, 2020

Policy expressions for string matching operations on a large set of string patterns tend to become long and complex. Resources consumed by the evaluation of such complex expressions are significant in terms of processing cycles, memory, and configuration size. You can create simpler, less resource-intensive expressions by using pattern matching.

Depending on the type of patterns that you want to match, you can use one of the following features to implement pattern matching:

- A pattern set is an array of indexed patterns used for string matching during default syntax policy evaluation. Example of a pattern set: imagetypes {svg, bmp, png, gif, tiff, jpg}.
- A data set is a specialized form of pattern set. It is an array of patterns of types number (integer), IPv4 address, or IPv6 address.

In many cases, you can use either pattern sets or data sets. However, in cases where you want specific matches for numerical data or IPv4 and IPv6 addresses, you must use data sets.

> **Note**:
> Pattern sets
> and data sets can be used only in default syntax policies.

To use pattern sets or data sets, first create the pattern set or data set and bind patterns to it. Then, when you configure a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

How string matching works with pattern sets and data sets

September 21, 2020

A pattern set or data set contains a set of patterns, and each pattern is assigned a unique index. When a policy is applied to a packet, an expression identifies a string to be evaluated, and the operator compares the string to the patterns defined in the pattern set or data set until a match is found or all patterns have been compared. Then, depending on its function, the operator returns either a boolean value that indicates whether or not a matching pattern was found or the index of the pattern that matches the string.

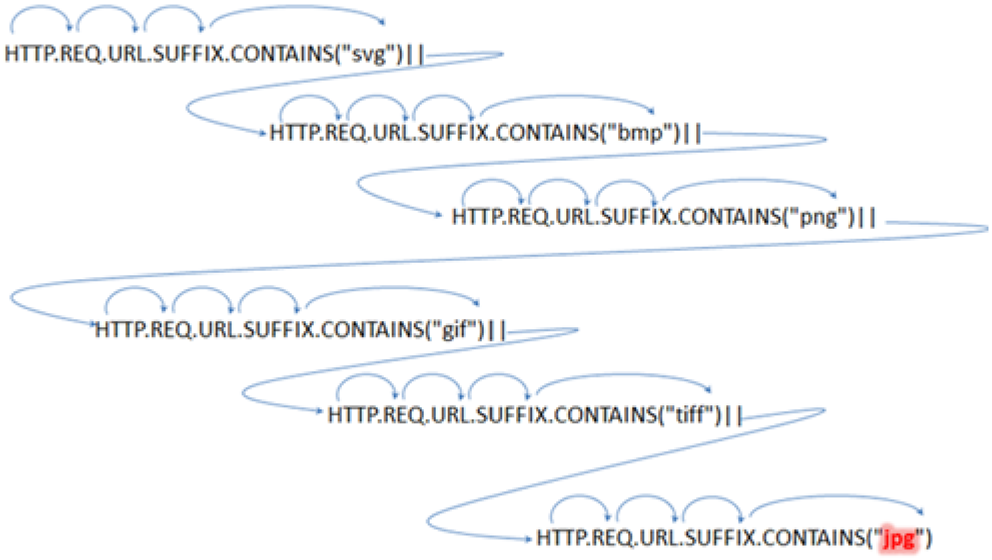
Note: This topic explains the working of a pattern set. Data sets work the same way. The only difference between pattern sets and data sets is the type of patterns defined in the set.

Consider the following use case to understand how patterns can be used for string matching.

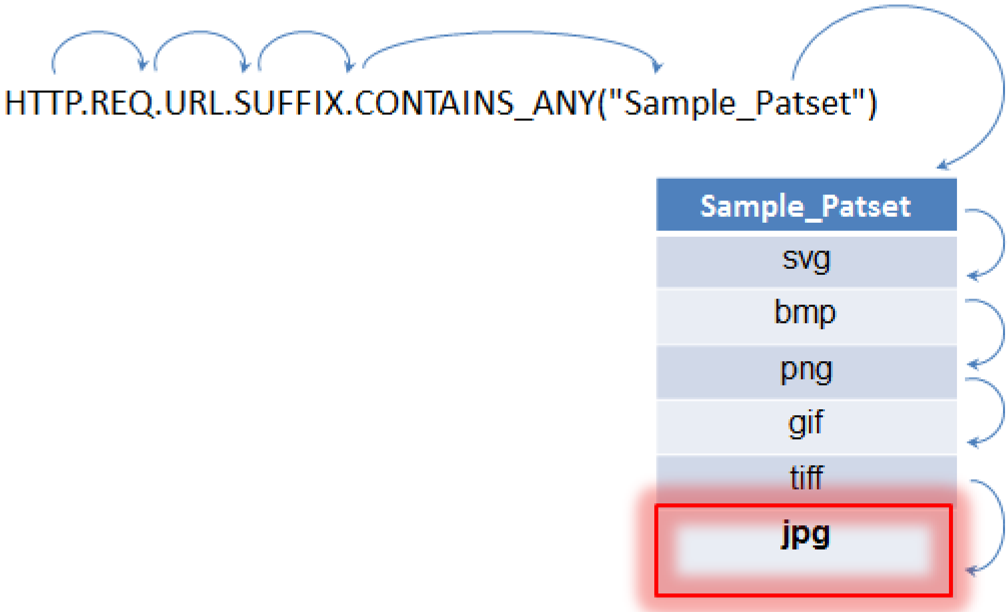
You want to determine whether the URL suffix (target text) contains any of the image file extensions. Without using pattern sets, you would have to define a complex expression, as follows:

```
HTTP.REQ.URL.SUFFIX.CONTAINS("svg") || HTTP.REQ.URL.SUFFIX.CONTAINS("
    bmp") || HTTP.REQ.URL.SUFFIX.CONTAINS("png") ||
HTTP.REQ.URL.SUFFIX.CONTAINS("gif") || HTTP.REQ.URL.SUFFIX.CONTAINS("
    tiff") || HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")
<!--NeedCopy-->
```

If the URL has a suffix of "jpg," with the above compound expression, the Citrix ADC appliance has to iterate through the entire compound expression sequentially, from one sub expression to the next, to determine that the request refers to a jpg image. The following figure shows the steps in the process.


HTTP.REQ.URL.SUFFIX.CONTAINS("svg")||
HTTP.REQ.URL.SUFFIX.CONTAINS("bmp")||
HTTP.REQ.URL.SUFFIX.CONTAINS("png")||
HTTP.REQ.URL.SUFFIX.CONTAINS("gif")||
HTTP.REQ.URL.SUFFIX.CONTAINS("tiff")||
HTTP.REQ.URL.SUFFIX.CONTAINS("jpg")

When a compound expression includes hundreds of sub expressions, the above process is resource intensive. A better alternative is an expression that invokes a pattern set, as shown in the following figure.



During policy evaluation as shown above, the operator (CONTAINS_ANY) compares the string identified in the request with the patterns defined in the pattern set until a match is found. With the Sample_Patset expression, the multiple iterations through six sub expressions are reduced to just one.

By eliminating the need to configure compound expressions that perform string matching with multiple OR operations, pattern sets or data sets simplify configuration and accelerate processing of requests and responses.

Configuring a Pattern Set

September 21, 2020

To configure a pattern set, you must specify the strings that are to serve as patterns. You can manually assign a unique index value to each of these patterns, or you can allow the index values to be assigned automatically.

Note: Pattern sets are case sensitive (unless you specify the expression to ignore case). Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

Points to remember about index values:

- You cannot bind the same index value to more than one pattern.

- An automatically assigned index value is one number larger than the highest index value of the existing patterns within the pattern set. For example, if the highest index value of existing patterns in a pattern set is 104, the next automatically assigned index value will be 105.
- If you do not specify an index for the first pattern, index value 1 is automatically assigned to that pattern.
- Index values are not regenerated automatically if one or more patterns are deleted or modified. For example, if the set contains five patterns, with indexes from 1 through 5, and if the pattern with an index of 3 is deleted, the other index values in the pattern set are not automatically regenerated to produce values from 1 through 4.
- The maximum index value that can be assigned to a pattern is 4294967290. If that value is already assigned to a pattern in the set, you must manually assign index values to any newly added patterns. An unused index value that is lower than a currently used value cannot be assigned automatically.

To configure a pattern set by using the command line interface

At the command prompt, do the following:

1. Create a pattern set.

   ```
   add policy patset <name>
   ```

   Example:

   ```
   > add policy patset samplepatset
   <!--NeedCopy-->
   ```

2. Bind patterns to the pattern set.

   ```
   bind policy patset <name> <string> [-index <positive_integer>]
   ```

   Example:

   ```
   > bind policy patset samplepatset product1 -index 1
   <!--NeedCopy-->
   ```

   Note: Repeat this step for all the patterns you want to bind to the pattern set.

3. Verify the configuration.

   ```
   show policy patset <name>
   ```

To configure a pattern set by using the configuration utility

1. Navigate to **AppExpert** > **Pattern Sets**.
2. In the details pane, click **Add** to open the **Create Pattern Set** dialog box.

3. Specify a name for the pattern set in the Name text box.
4. Under Specify Pattern, type the first pattern and, optionally, specify values for the following parameters:
    - Treat back slash as escape character—Select this check box to specify that any backslash characters that you might include in the pattern are to be treated as escape characters.
    - Index—A user assigned index value, from 1 through 4294967290.
5. Verify that you have entered the correct characters, and then click **Add**.
6. Repeat steps 4 and 5 to add additional patterns, and then click **Create**.

Configuring a data set

September 21, 2020

To configure a data set, you must specify the strings that are to serve as patterns, and assign a type (number, IPv4 address, or IPv6 address) to each pattern. You can manually assign a unique index value to each of these patterns, or you can allow the index values to be assigned automatically. Dataset is not related to HTTP or any 7 layer protocol. It works only on text/string. There are different types of dataset such as num, ulong, ipv4, and ipv6. To use a dataset on IP, we have to convert the input into text/string.

Note: Data sets are case sensitive (unless you specify the expression to ignore case). Therefore, the string pattern "product1," for example, is not the same as the string pattern "Product1."

The rules applied for index values of data sets are the same as those applied for pattern sets. For information about index values, see Configuring a Pattern Set.

To configure a data set by using the command line interface

At the command prompt, do the following:

1. Create a data set.

```
add policy dataset <name> <type>
```

Example:

```
```
> add policy dataset sampledataset ipv4
<!--NeedCopy--> ```
```

1. Bind patterns to the data set.

    ```
    bind policy dataset <name> <value> [-index <positive_integer>]
    ```
    ```

Example:

```
```
> bind policy dataset sampledataset 10.102.29.1 -index 1
<!--NeedCopy--> ```
```

> **Note**:
>
> Repeat this step for all the patterns you want to bind to the data set.

1. add policy expression sampleexpression client.ip.src.typecast_text_t.equals_any("sampledataset")
2. Verify the configuration.

   `show policy dataset <name>`

### To configure a data set by using the configuration utility

Navigate to **AppExpert** > **Data Sets**, click **Add** and specify the relevant details.

## Using pattern sets and data sets

September 21, 2020

Default syntax policy expressions that take pattern sets or data sets as an argument can be used to perform string matching operations.

The usage is as follows:

```
<text>.<operator>("<name>")
<!--NeedCopy-->
```

where,

- `<text>` is the expression that identifies a string in a packet. Example: HTTP.REQ.HEADER("Host").
- `<operator>` is one of the operators described in the Pattern Set Types table pdf.

For sample usage, see Sample Usage.

Sample usage

September 21, 2020
```

To understand the usage of pattern sets in expressions, consider the example of a pattern set named “imagetypes.”

Patterns	Index value
svg	1
bmp	2
png	3
gif	4
tiff	5
jpg	6

Table 1. Pattern set "imagetypes"

Example 1: Determine whether the suffix of an HTTP request is one of the file extensions defined in the "imagetypes" pattern set.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_ANY("imagetypes")
- **Sample URL.** `http://www.example.com/homepageicon.jpg`
- **Result.** TRUE

Example 2: Determine whether the suffix of an HTTP request is one of the file extensions defined in the "imagetypes" pattern set, and return the index of that pattern.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes")
- **Sample URL.** `http://www.example.com/mylogo.png`
- **Result.** 4 (The index value of the pattern "gif".)

Example 3: Use the index value of a pattern to determine whether the URL suffix is within a specified index-value range.

- **Expression.** HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(3) && HTTP.REQ.URL.SUFFIX.EQUALS
- **Sample URL.** `http://www.example.com/mylogo.png`
- **Result.** TRUE (The index value of gif file types is 4.)

Example 4: Implement one set of policies for file extensions bmp, jpg, and png, and a different set of policies for gif, tiff, and svg files.

An expression that returns the index of a matched pattern can be used to define traffic subsets for a web application. The following two expressions could be used in content switching policies for a content switching virtual server:

- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").LE(3)
- HTTP.REQ.URL.SUFFIX.EQUALS_INDEX("imagetypes").GE(4)

Variables

September 21, 2020

Variables are named objects that store information in the form of tokens. These tokens are used within and across different transactions on the NetScaler Appliance for internal computation and policy processing.

The NetScaler appliance supports creation of variables of the following types:

- **Singleton variables.** Can have a single value of one of the following types: ulong and text (max-size). The ulong type is an unsigned 64-bit integer, the text type is a sequence of bytes, and max-size is the maximum number of bytes in the sequence.
- **Map variables.** Maps hold values associated with keys: each key-value pair is called a map entry. The key for each entry is unique within the map. Maps are specified as follows:

  map (key_type, value_type, max-values).

  where,

  - *key_type* is the data type of the key. It is of type text (max-size).
  -
    ```
    *value_type* is the data type of the values of the map. It can
        be of type ulong or text (max-size).
    ```

    ```
    *max-values* is the maximum number of entries that the map can
        contain. It is of type ulong.
    ```

Values for these variables are set using assignments which must be invoked on policy actions.

> - **Note**: Variables are not yet supported in a high-availability setup or in a cluster.

Variables Scope

A map variable or a singleton variable can have a global scope. Alternatively, the scope of a singleton variable can be limited to a single transaction.

- **Global Scope Variable** - A variable with global scope (the default) has only one instance, and that instance has the same value(s) across all cores of a NetScaler appliance and across all nodes of a cluster or HA configuration. Global variable values exist until they are explicitly deleted, until they expire, or until a standalone appliance is restarted or all nodes of a cluster or HA configuration are restarted.

- **Transaction Scope Variable** - A variable with transaction scope has a separate instance, with its own value, for each transaction processed by the NetScaler appliance. When the transaction processing is complete, the transaction variable value is deleted.

> **Note**: Transaction scope variables are available in NetScaler release 10.5.e or later.

Configuring and using Variables

September 21, 2020

You must first create a variable and then assign a value or specify the operation that must be performed on the variable. After performing these operations, you can use the assignment as a policy action.

Note: Once configured, a variable's settings cannot be modified or reset. If the variable needs to be changed, the variable and all references to the variable (expressions and assignments) must be deleted. The variable can then be re-added with new settings, and the references (expressions and assignments) can be re-added.

To configure variables by using the command line interface

1. Create a variable.

```
add ns variable <name> -type <string> [-scope global] [-ifFull ( undef
    | lru )] [-ifValueTooBig ( undef | truncate )] [-ifNoValue ( undef |
     init )] [-init <string>] [-expires <positive_integer>] [-comment <
    string>]
<!--NeedCopy-->
```

Note: Refer to the man page "man add ns variable" for description of the command parameters.

Example 1: Create a ulong variable named "my_counter" and initialize it to 1.

```
add ns variable my_counter  - type ulong -init 1
<!--NeedCopy-->
```

Example 2: Create a map named "user_privilege_map". The map will contain keys of maximum length 15 characters and text values of maximum length 10 characters, with a maximum of 10000 entries.

```
add ns variable user_privilege_map -type map(text(15),text(10),10000)
<!--NeedCopy-->
```

Note: If the map contains 10000 unexpired entries, assignments for new keys reuse one of the least recently used entries. By default, an expression trying to get a value for a non-existent key will initialize an empty text value.

Assign the value or specify the operation to be performed on the variable. This is done by creating an assignment.

```
add ns assignment <name> -variable <expression> [-set <expression> | -
    add <expression> | -sub <expression> | -append <expression> | -clear
    ] [-comment <string>]
<!--NeedCopy-->
```

Note: A variable is referenced by using the variable selector ($). Therefore,
$variable1 is used to refer to text or ulong variables. Similarly,
$variable2[key-expression] is used to refer to map variables.

Example 1: Define an assignment named "inc_my_counter" that automatically adds 1 to the "my_counter" variable.

```
add ns assignment inc_my_counter -variable $my_counter -add 1
<!--NeedCopy-->
```

Example 2: Define an assignment named "set_user_privilege" that adds to the "user_privilege_map" variable an entry for the client's IP address with the value returned by the "get_user_privilege" HTTP callout.

```
add ns assignment set_user_privilege -variable $user_privilege_map[
    client.ip.src.typecast_text_t] -set sys.http.callout(
    get_user_privilege)
<!--NeedCopy-->
```

Note: If an entry for that key already exists, the value will be replaced. Otherwise a new entry for the key and value will be added. Based on the previous declaration for user_privilege_map, if the map already has 10000 entries, one of the least recently used entries will be reused for the new key and value.

1. Invoke the variable assignment in a policy.

   There are two functions that can operate on map variables.

   - **$name.valueExists(key-expression).** Returns true if there is a value in the map selected by the key-expression. Otherwise returns false. This function will update the expiration and LRU information if the map entry exists, but will not create a new map entry if the value does not exist.

- **$name.valueCount.** Returns the number of values currently held by the variable. This is the number of entries in a map. For a singleton variable, this is 0 if the variable is uninitialized or 1 otherwise.

Example: Invoke the assignment named "set_user_privilege" with a compression policy.

```
add cmp policy set_user_privilege_pol -rule $user_privilege_map.
    valueExists(client.ip.src.typecast_text_t).not -resAction
    set_user_privilege
<!--NeedCopy-->
```

Use Case to insert HTTP header in the response side

The following example shows an example of a singleton variable.

Add a singleton variable of type text. This variable can hold maximum 100 Bytes data.

```
add ns variable http_req_data -type text(100) -scope transaction
<!--NeedCopy-->
```

Add an assignment action, which will be used to store the HTTP request data into the variable.

```
add ns assignment set_http_req_data -variable $http_req_data -set http.
    req.body(100)
<!--NeedCopy-->
```

Add a rewrite action to insert HTTP header, whose value will be fetched from the variable.

```
add rewrite action act_ins_header insert_http_header user_name
    $http_req_data.after_str("user_name").before_str("password")
<!--NeedCopy-->
```

Add a rewrite policy which will evaluate in the request time, and take assignment action to store data. When we hit this policy, we'll take assignment action and store the data into the ns variable (http_req_data)

```
add rewrite policy pol_set_variable true set_http_req_data

bind rewrite global pol_set_variable 10 -type req_dEFAULT
<!--NeedCopy-->
```

Add a rewrite policy which will evaluate in the response time, and add an HTTP header in the response.

```
add rewrite policy pol_ins_header true act_ins_header

```

```
bind rewrite global pol_ins_header 10 -type res_dEFAULT
<!--NeedCopy-->
```

To configure variables by using the GUI

1. Navigate to **AppExpert** > **NS Variables**, to create a variable.
2. Navigate to **AppExpert** > **NS Assignments,** to assign value(s) to the variable.
3. Navigate to the appropriate feature area where you want to configure the assignment as an action.

Use Case: Caching User Privileges

September 21, 2020

In this use case, user privileges ("GOLD", "SILVER", and so on) must be retrieved from an external web service.

To achieve this use case, perform the following operations

Create an HTTP callout to fetch the user privileges from the external web service.

```
add policy httpcallout <name> [-IPAddress <ip_addr|ipv6_addr>] [-port <
    port>] [-vServer <string>] [-returnType <returnType>] [-httpMethod (
    GET | POST )] [-hostExpr <string>] [-urlStemExpr <string>] [-headers
     <name(value)> ...] [-parameters <name(value)> ...] [-bodyExpr <
    string>][-fullReqExpr <string>] [-scheme ( http | https )] [-
    resultExpr <string>] [-cacheForSecs <secs>] [-comment <string>]

add policy httpcallout get_user_privilege -ipaddress 10.217.193.84 -
    port 80 -returnType text -httpMethod GET -hostExpr '"/
    get_user_privilege"' -resultExpr 'http.res.body(5)'
<!--NeedCopy-->
```

Store the privileges in a variable.

```
add ns variable <name> -type <string> [-scope ( global | transaction )
    ][-ifFull ( undef | lru )] [-ifValueTooBig ( undef | truncate )][-
    ifNoValue ( undef | init )] [-init <string>] [-expires <
    positive_integer>] [-comment <string>]

```

```
add ns variable user_privilege_map -type map(text(15),text(10),10000) -expires 1200

add ns assignment set_user_privilege -variable $user_privilege_map[client.ip.src] -set sys.http_callout(get_user_privilege)
<!--NeedCopy-->
```

Create a policy to check if there is already a cached entry for the client’s IP address; if not, it calls the HTTP callout to set a map entry for the client.

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy set_user_privilege_pol -rule $user_privilege_map.valueExists(client.ip.src).not -resAction set_user_privilege>
<!--NeedCopy-->
```

Create a policy that compresses if the cached privilege entry for the client is “GOLD”.

```
add cmp policy <name> -rule <expression> -resAction <string>

add cmp policy compress_if_gold_privilege_pol -rule '$user_privilege_map[client.ip.src].eq("GOLD")' -resAction compress
<!--NeedCopy-->
```

Bind the compression policies globally.

```
bind cmp global <policyName> [-priority <positive_integer>] [-state ( ENABLED | DISABLED )] [-gotoPriorityExpression <expression>] [-type <type>] [-invoke (<labelType> <labelName>) ]

bind cmp global set_user_privilege_pol -priority 10 NEXT

bind cmp global compress_if_gold_privilege_pol -priority 20 END
<!--NeedCopy-->
```

Use Case: Limiting the Number of Sessions

September 21, 2020

In this use case, the requirement is to limit the number of active backend sessions. In the deployment, each session login has login in the URL and each session logout has logout in the URL. On successful login, the backend sets a sessionid cookie with a unique 10 character value.

To achieve this use case, perform the following operations:

1. Create a map variable that can store each active session. The key of the map is the sessionid. The expiry time for the variable is set to 600 seconds (10 minutes).

   ```
   > add ns variable session_map -type map(text(10),ulong,100) -
     expires 600
   <!--NeedCopy-->
   ```

2. Create the following assignments for the map variable:

   - Create an entry for the sessionid and set that value to 1 (this value is not actually used).

     ```
     > add ns assignment add_session -variable '$session_map[http.
       req.cookie.value("sessionid")]' -set 1
     <!--NeedCopy-->
     ```

   - Deallocate the entry for a session ID, which implicitly decrements the value count for session_map.

     ```
     > add ns assignment delete_session -variable '$session_map[
       http.req.cookie.value("sessionid")]' -clear
     <!--NeedCopy-->
     ```

3. Create responder policies for the following:

   - To check if a map entry exists for that sessionid in the HTTP request. The add_session assignment is executed if the map entry does not exist.

     ```
     > add responder policy add_session_pol 'http.req.url.contains
       ("twbkwbis.P_SabanciLogin") || $session_map.valueExists(
       http.req.cookie.value("netsuis"))' add_session
     <!--NeedCopy-->
     ```

     Note: The
     valueExists() function in the
     add_session_pol policy counts as a reference to the session’s map entry, so each request resets the expiration timeout for its session. If no requests for a session are received after 10 minutes, the session’s entry will be deallocated.

   - To check when the session is logged out. The delete_session assignment is executed.

     ```
     add responder policy delete_session_pol "http.req.url.
       contains(\"Logout\")" delete_session
     ```

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

- To check for login requests and if the number of active sessions exceed 100. If these conditions are satisfied, in order to limit the number of sessions, the user is redirected to a page that indicates that the server is busy.

  ```
  add responder action redirect_too_busy redirect "/too_busy.
     html"
  add responder policy check_login_pol "http.req.url.contains
     (\"twbkwbis.P_SabanciLogin\") && $session_map.valueCount >
      100" redirect_too_busy
  <!--NeedCopy-->
  ```

4. Bind the responder policies globally.

   ```
   bind responder global add_session_pol 30 next
   bind responder global delete_session_pol 10
   bind responder global check_login_pol 20
   <!--NeedCopy-->
   ```

Policies and expressions

September 21, 2020

The following topics provide the conceptual and reference information that you require for configuring advanced policies on the Citrix® NetScaler® appliance.

To know about all the advanced policy expressions supported on the Citrix ADC appliance, see Policy Expressions

Introduction to Policies and Expressions	Describes the purpose of expressions, policies, and actions, and how different Citrix ADC applications make use of them.

Configuring Advanced Policies	Describes the structure of advanced policies and how to configure them individually and as policy banks.
Configuring Advanced Expressions: Getting Started	Describes expression syntax and semantics, and briefly introduces how to configure expressions and policies.
Advanced Expressions: Evaluating Text	Describes expressions that you configure when you want to operate on text (for example, the body of an HTTP POST request or the contents of a user certificate).

Advanced Expressions: Working with Dates, Times, and Numbers	Describes expressions that you configure when you want to operate on any type of numeric data (for example, the length of a URL, a client's IP address, or the date and time that an HTTP request was sent).
Advanced Expressions: Parsing HTTP, TCP, and UDP Data	Describes expressions for parsing IP and IPv6 addresses, MAC addresses, and data that is specific to HTTP and TCP traffic.

Advanced Expressions: Parsing SSL Certificates	Describes how to configure expressions for SSL traffic and client certificates, for example, how to retrieve the expiration date of a certificate or the certificate issuer.				
Advanced Expressions: IP and MAC Addresses, Throughput, VLAN IDs	Describes expressions that you can use to work with any other client- or server-related data not discussed in other chapters.	Typecasting Data	Describes expressions for transforming data of one type to another.	Regular Expressions	Describes how to pass regular expressions as arguments to operators in advanced expressions.
Configuring Classic Policies and Expressions	Provides details on how to configure the simpler policies and expressions known as classic policies and classic expressions.	Expressions Reference	A reference for classic and advanced expression arguments.	Summary Examples of Advanced Expressions and Policies	Examples of classic and advanced expressions and policies, in both quick reference and tutorial format, that you can customize for your own use.

Tutorial Examples of Advanced Policies for Rewrite	Examples of advanced policies for use in the Rewrite feature.
Tutorial Examples of Classic Policies	Examples of classic policies for Citrix ADC features such as application firewall and SSL.
Migration of Apache mod_rewrite Rules to Advanced Policies	Examples of functions that were written using the Apache HTTP Server mod_rewrite engine, with examples of these functions after translation into Rewrite and Responder policies on the Citrix ADC.

Introduction to policies and expressions

September 21, 2020

For many NetScaler features, policies control how a feature evaluates data, which ultimately determines what the feature does with the data. A policy uses a logical expression, also called a rule, to evaluate requests, responses, or other data, and applies one or more actions determined by the outcome of the evaluation. Alternatively, a policy can apply a profile, which defines a complex action.

Some NetScaler features use default syntax policies, which provide greater capabilities than do the older, classic, policies. If you migrated to a newer release of the NetScaler software and have configured classic policies for features that now use default syntax policies, you might have to manually migrate policies to the default syntax.

Classic and advanced policies

September 21, 2020

> **Warning**
>
> Classic policy expressions are no longer supported from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies

Classic policies evaluate basic characteristics of traffic and other data. For example, classic policies can identify whether an HTTP request or response contains a particular type of header or URL.

Advanced policies can perform the same type of evaluations as classic policies. In addition, default syntax policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, transforming data in the body of a request into an HTTP header).

In addition to assigning a policy an action or profile, you bind the policy to a particular point in the processing associated with the NetScaler features. The bind point is one factor that determines when the policy will be evaluated.

Benefits of using advanced policies

Default syntax policies use a powerful expression language that is built on a class-object model, and they offer several options that enhance your ability to configure the behavior of various NetScaler features. With default syntax policies, you can do the following:

- Perform fine-grained analyses of network traffic from layers 2 through 7.

- Evaluate any part of the header or body of an HTTP or HTTPS request or response.
- Bind policies to the multiple bind points that the default syntax policy infrastructure supports at the default, override, and virtual server levels.
- Use goto expressions to transfer control to other policies and bind points, as determined by the result of expression evaluation.
- Use special tools such as pattern sets, policy labels, rate limit identifiers, and HTTP callouts, which enable you to configure policies effectively for complex use cases.

Additionally, the GUI extends robust graphical user interface support for default syntax policies and expressions and enables users who have limited knowledge of networking protocols to configure policies quickly and easily. The GUI also includes a policy evaluation feature for default syntax policies. You can use this feature to evaluate a default syntax policy and test its behavior before you commit it, thus reducing the risk of configuration errors.

Basic components of an advanced policy

Following are a few characteristics of an Advanced policy:

- Name.

  Each policy has a unique name.

- Rule.

  The rule is a logical expression that enables the NetScaler feature to evaluate a piece of traffic or another object.

  For example, a rule can enable the NetScaler to determine whether an HTTP request originated from a particular IP address, or whether a Cache-Control header in an HTTP request has the value "No-Cache."

  Advanced policies can use all of the expressions that are available in a classic policy, with the exception of classic expressions for the SSL VPN client. In addition, Advanced policies enable you to configure more complex expressions.

- Bindings.

  To ensure that the NetScaler can invoke a policy when it is needed, you associate the policy, or bind it, to one or more bind points.

  You can bind a policy globally or to a virtual server. For more information, see "About policy bindings."

- An associated action.

  An action is a separate entity from a policy. Policy evaluation ultimately results in the NetScaler performing an action.

For example, a policy in the integrated cache can identify HTTP requests for .png or .jpeg files. An action that you associate with this policy determines that the responses to these types of requests are served from the cache.

For some features, you configure actions as part of a more complex set of instructions known as a profile.

How different NetScaler features use policies

The NetScaler supports a variety of features that rely on policies for operation. The following table summarizes how the NetScaler features use policies.

Feature Name	Policy Type	How You Use Policies in the Feature
System	Classic	For the Authentication function, policies contain authentication schemes for different authentication methods. For example, you can configure LDAP and certificate-based authentication schemes. You also configure policies in the Auditing function.
DNS	Advanced	To determine how to perform DNS resolution for requests.
SSL	Classic and Advanced	To determine when to apply an encryption function and add certificate information to clear text. To provide end-to-end security, after a message is decrypted, the SSL feature re-encrypts clear text and uses SSL to communicate with Web servers.
Compression	Classic and Advanced	To determine what type of traffic is compressed.
Integrated Caching	Advanced	To determine whether HTTP responses are cacheable.

Feature Name	Policy Type	How You Use Policies in the Feature
Responder	Advanced	To configure the behavior of the Responder function.
Protection Features	Classic	To configure the behavior of the Filter, SureConnect, and Priority Queuing functions.
Content Switching	Classic and Advanced	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.
AAA - Traffic Management	Classic. Exceptions: Traffic policies support only default syntax policies aand authorization policies support both classic and default syntax policies.	To check for client-side security before users log in and establish a session. Traffic policies, which determine whether single sign-on (SSO) is required, use only the default syntax. Authorization policies authorize users and groups that access intranet resources through the appliance.
Cache Redirection	Classic	To determine whether responses are served from a cache or from an origin server.

Feature Name	Policy Type	How You Use Policies in the Feature
Rewrite	Advanced	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a new home page, or a new server, or a selected server based on the address of the incoming request, or you can modify the data to mask server information in a response for security purposes. The URL Transformer function identifies URLs in HTTP transactions and text files for the purpose of evaluating whether a URL should be transformed.
Application Firewall	Classic and Advanced	To identify characteristics of traffic and data that should or should not be admitted through the firewall.
NetScaler Gateway, Clientless Access function	Advanced	To define rewrite rules for general Web access using the NetScaler Gateway.
NetScaler Gateway	Classic	To determine how the NetScaler Gateway performs authentication, authorization, auditing, and other functions.

About actions and profiles

Policies do not themselves take action on data. Policies provide read-only logic for evaluating traffic. To enable a feature to perform an operation based on a policy evaluation, you configure actions or profiles and associate them with policies.

Note: Actions and profiles are specific to particular features. For information about assigning actions and profiles to features, see the documentation for the individual features.

About actions

Actions are steps that the NetScaler takes, depending on the evaluation of the expression in the policy. For example, if an expression in a policy matches a particular source IP address in a request, the action that is associated with this policy determines whether the connection is permitted.

The types of actions that the NetScaler can take are feature specific. For example, in Rewrite, actions can replace text in a request, change the destination URL for a request, and so on. In Integrated Caching, actions determine whether HTTP responses are served from the cache or an origin server.

In some NetScaler features actions are predefined, and in others they are configurable. In some cases, (for example, Rewrite), you configure the actions using the same types of expressions that you use to configure the associated policy rule.

About profiles

Some NetScaler features enable you to associate profiles, or both actions and profiles, with a policy. A profile is a collection of settings that enable the feature to perform a complex function. For example, in the application firewall, a profile for XML data can perform multiple screening operations, such as examining the data for illegal XML syntax or evidence of SQL injection.

Use of actions and profiles in particular features

The following table summarizes the use of actions and profiles in different NetScaler features. The table is not exhaustive. For more information about specific uses of actions and profiles for a feature, see the documentation for the feature.

Feature	Use of an Action	Use of a Profile
Application firewall	Synonymous with a profile	All application firewall features use profiles to define complex behaviors, including pattern-based learning. You add these profiles to policies.
NetScaler Gateway	The following features of the NetScaler Gateway use actions: Pre-Authentication. Uses Allow and Deny actions. You add these actions to a profile., Authorization. Uses Allow and Deny actions. You add these actions to a policy. TCP Compression. Uses various actions. You add these actions to a policy.	The following features use a profile: Pre-Authentication, Session, Traffic, and Clientless Access. After configuring the profiles, you add them to policies.
Rewrite	You configure URL rewrite actions and add them to a policy.	Not used.
Integrated Caching	You configure caching and invalidation actions within a policy	Not used.
AAA - Traffic Management	You select an authentication type, set an authorization action of ALLOW or DENY, or set auditing to SYSLOG or NSLOG.	You can configure session profiles with a default timeout and authorization action.
Protection Features	You configure actions within policies for the following functions: Filter, Compression, Responder, and SureConnect.	Not used.
SSL	You configure actions within SSL policies	Not used.

Feature	Use of an Action	Use of a Profile
System	The action is implied. For the Authentication function, it is either Allow or Deny. For Auditing, it is Auditing On or Auditing Off.	Not used.
DNS	The action is implied. It is either Drop Packets or the location of a DNS server.	Not used.
SSL Offload	The action is implied. It is based on a policy that you associate with an SSL virtual server or a service.	Not used.
Compression	Determine the type of compression to apply to the data	Not used.
Content Switching	The action is implied. If a request matches the policy, the request is directed to the virtual server associated with the policy.	Not used.
Cache Redirection	The action is implied. If a request matches the policy, the request is directed to the origin server.	Not used.

About policy bindings

A policy is associated with, or bound to, an entity that enables the policy to be invoked. For example, you can bind a policy to request-time evaluation that applies to all virtual servers. A collection of policies that are bound to a particular bind point constitutes a policy bank.

Following is an overview of different types of bind points for a policy:

- Request time global.

  A policy can be available to all components in a feature at request time.

- Response time global.

A policy can be available to all components in a feature at response time.

- Request time, virtual server-specific.

  A policy can be bound to request-time processing for a particular virtual server. For example, you can bind a request-time policy to a cache redirection virtual server to ensure that particular requests are forwarded to a load balancing virtual server for the cache, and other requests are sent to a load balancing virtual server for the origin.

- Response time, virtual server-specific.

  A policy can also be bound to response-time processing for a particular virtual server.

- User-defined policy label.

  For default syntax policies, you can configure custom groupings of policies (policy banks) by defining a policy label and collecting a set of related policies under the policy label.

- Other bind points.

  The availability of additional bind points depends on type of policy (classic or default syntax), and specifics of the relevant NetScaler feature. For example, classic policies that you configure for the
  NetScaler Gateway have user and group bind points.

For additional information about default syntax policy bindings, see "Bind policies that use the default syntax" and Configure a policy bank for a virtual server. For additional information about classic policy bindings, see "Configure a classic policy."

About evaluation order of policies

For classic policies, policy groups and policies within a group are evaluated in a particular order, depending on the following:

- The bind point for the policy, for example, whether the policy is bound to request-time processing for a virtual server or global response-time processing. For example, at request time, the NetScaler evaluates all request-time classic policies before evaluating any virtual server-specific policies.
- The priority level for the policy. For each point in the evaluation process, a priority level that is assigned to a policy determines the order of evaluation relative to other policies that share the same bind point. For example, when the NetScaler evaluates a bank of request-time, virtual server-specific policies, it starts with the policy that is assigned to the lowest priority value. In classic policies, priority levels must be unique across all bind points.

For Advanced policies, as with classic policies, the NetScaler selects a grouping, or bank, of policies at a particular point in overall processing. Following is the order of evaluation of the basic groupings, or banks, of Advanced policies:

1. Request-time global override
2. Request-time, virtual server-specific (one bind point per virtual server)
3. Request-time global default
4. Response-time global override
5. Response-time virtual server-specific
6. Response-time global default

However, within any of the preceding banks of policies, the order of evaluation is more flexible than in classic policies. Within a policy bank, you can point to the next policy to be evaluated regardless of the priority level, and you can invoke policy banks that belong to other bind points and user-defined policy banks.

Order of evaluation based on traffic flow

As traffic flows through the NetScaler and is processed by various features, each feature performs policy evaluation. Whenever a policy matches the traffic, the NetScaler stores the action and continues processing until the data is about to leave the NetScaler. At that point, the NetScaler typically applies all matching actions. Integrated Caching, which only applies a final Cache or NoCache action, is an exception.

Some policies affect the outcome of other policies. Following are examples:

- If a response is served from the integrated cache, some other NetScaler features do not process the response or the request that initiated it.
- If the Content Filtering feature prevents a response from being served, no subsequent features evaluate the response.

If the application firewall rejects an incoming request, no other features can process it.

Classic and advanced policy expressions

September 21, 2020

One of the most fundamental components of a policy is its rule. A policy rule is a logical expression that enables the policy to analyze traffic. Most of the policy's functionality is derived from its expression.

An expression matches characteristics of traffic or other data with one or more parameters and values. For example, an expression can enable the NetScaler to accomplish the following:

- Determine whether a request contains a certificate.
- Determine the IP address of a client that sent a TCP request.

- Identify the data that an HTTP request contains (for example, a popular spreadsheet or word processing application).
- Calculate the length of an HTTP request.

About classic expressions

Classic expressions enable you to evaluate basic characteristics of data. They have a structured syntax that performs string matching and other operations.

Following are a few simple examples of classic expressions:

- An HTTP response contains a particular type of Cache Control header.

res.http.header Cache-Control contains public

- An HTTP response contains image data.

res.http.header Content-Type contains image/

- An SSL request contains a certificate.

req.ssl.client.cert exists

About advanced policy expressions

Any feature that uses default syntax policies also uses Advanced expressions. For information about which features use Advanced policies, see the table NetScaler Feature, Policy Type, and Policy Usage.

Advanced policy expressions have a few other uses. In addition to configuring Advanced expressions in policy rules, you configure Advanced
expressions in the following situations:

- Integrated Caching:

  You use Advanced policy expressions to configure a selector for a content group in the integrated cache.

- Load Balancing:

  You use Advanced policy expressions to configure token extraction for a load balancing virtual server that uses the TOKEN method for load balancing.

- Rewrite:

  You use Advanced policy expressions to configure rewrite actions.

- Rate-based policies:

  You use Advanced policy expressions to configure limit selectors when configuring a policy to control the rate of traffic to various servers.

Following are a few simple examples of Advanced policy expressions:

- An HTTP request URL contains no more than 500 characters.

```
http.req.url.length \<= 500
```

- An HTTP request contains a cookie that has fewer than 500 characters.

```
http.req.cookie.length \< 500
```

- An HTTP request URL contains a particular text string.

```
http.req.url.contains(".html")
```

Converting policy expressions using NSPEPI tool

September 21, 2020

You can convert a classic expression to the Advanced policy expression by using the nspepi conversion tool. You can also use the tool to convert all the classic expressions in the NetScaler configuration to the Advanced policy (with the exception of NetScaler entities that currently support only classic expressions).

The conversion tool does not convert policies configured for the following features because the features currently support only classic policies:

- Authentication, Pre-authentication
- SSL
- Cache redirection
- VPN (session, traffic, and tunnel traffic)
- Content filtering (The responder feature not only provides you with functionality that is equivalent to that provided by the content filtering feature but also surpasses the content filtering feature in the use cases that it supports. Additionally, responder supports the more powerful default syntax for policy expressions.)

The following NetScaler features support both classic and Advanced policy expression and, therefore, support the conversion of classic expressions to Advanced policy expressions:

- Application firewall policies
- Authorization policies
- Named expressions
- Compression policies
- Content switching policies
- User-defined, rule-based tokens/persistency (the –rule parameter value that is specified for a load balancing virtual server)

About the conversion process

When parsing a NetScaler configuration file, the conversion tool performs the following actions:

1. In commands that create classic named expressions, the conversion tool replaces the names of the classic expressions with Advanced policy expression.
2. In commands that support only the classic policy, if classic named expressions are used, the conversion tool replaces the names of the classic expressions with the actual classic expressions they represent. This action ensures that the names of expressions in classic-only features do not reference the Advanced policy expression created from Step 1.
3. In commands associated with entities that support both the classic syntax and the Advanced policies, the conversion tool replaces all classic expressions in commands with Advanced policy expressions.

Example:

Consider the following sample configuration commands:

```
add policy expression ne_c1 "METHOD == GET"
add policy expression ne_c2 "ne_c1 || URL == /*.htm "
add filter policy pol1 -rule "ne_c2" -reqAction YES
add cmp policy pol2 -rule "REQ.HTTP.HEADER Accept CONTAINS \'text/html\'" -resAction COMPRESS
add cmp policy pol3 -rule "ne_c1 || ne_c2" -resAction GZIP
<!--NeedCopy-->
```

In the commands that create the classic named expressions ne_c1 and ne_c2, the tool replaces the names of the expressions with actual Advanced policy expression. This action, which corresponds to Step 1 described earlier, results in the following commands:

```
add policy expression ne_c2 "HTTP.REQ.METHOD.EQ(\"GET\")"

add policy expression ne_c1 "HTTP.REQ.URL.SUFFIX.EQ(\"htm\")"
<!--NeedCopy-->
```

The filter policy command supports only the classic syntax. Therefore, the conversion tool replaces the classic named expression ne_c1 with the actual classic expression it represents. Note that the tool replaces ne_c1 in the expression for ne_c2, and then replaces ne_c2 in the filter policy with the classic expression. This action, which corresponds to Step 2 described earlier, results in the following command:

`add filter policy pol1 -rule "METHOD == GET || URL == /\*.htm"-reqAction YES`

The compression feature supports both classic and Advanced policy expression. Therefore, in the command that creates the compression policy pol2, the conversion tool replaces the expression with

Advanced policy expression. This action, which corresponds to Step 3 described earlier, results in the following command:

```
add cmp policy pol2 -rule "HTTP.REQ.HEADER(\"Accept\").AFTER_STR(\"text/
html\").LENGTH.GT(0)"-resAction COMPRESS
```

The command that creates the compression policy pol3 is unaffected by the conversion process because, after the conversion process is complete, ne_c1 and ne_c2 reference the Advanced policy expression that results from Step 1.

Client security messages are not supported in the newer default policy format and, therefore, are lost. The
SYS.EVAL_CLASSIC_EXPR function is replaced with a Advanced policy expression. The following entities support the
SYS.EVAL_CLASSIC_EXPR function:

- DNS policies
- Rate limit selectors
- Cache selectors
- Cache policies
- Content switching policies
- Rewrite policies
- URL transformation policies
- Responder policies
- Application firewall policies
- Authorization policies
- Compression policies
- CVPN access policies

After performing the conversion, the tool saves the changes in a new configuration file. The new configuration file is created in the directory in which the input file exists. The name of the new configuration file is the same as the name of the input configuration file except for the string new_ used as a prefix. Conversion warnings are reported in a warning line at the end of the screen output. Additionally, a warning file is created in the directory in which the input configuration file resides. For more information about the warning file and the types of warnings that are reported, see “Conversion warnings.”

Convert expressions

You can use the nspepi tool to convert a single classic expression to the Advanced policy expression. The nspepi tool must be run from the shell prompt on the NetScaler appliance.

Convert a classic expression to the Advanced policy by using the CLI

At the shell prompt, type:

‘nspepi -e “<classic expression>”

Example:

```
root@NS# nspepi -e "REQ.HTTP.URL == /*.htm"
 "HTTP.REQ.URL.REGEX_MATCH(re#/(.*)\.htm#)"
<!--NeedCopy-->
```

Convert a NetScaler configuration file

You can use the nspepi tool to convert all the classic expressions in a NetScaler configuration file to the Advanced policy (except for those commands that do not support the Advanced policy). The nspepi tool must be run from the shell prompt on the NetScaler appliance.

Convert all the classic expressions in a NetScaler configuration file to the default syntax by using the CLI

At the shell prompt, type:

nspepi -f “<ns config file>” -v

Example:

```
root@NS# nspepi -f ns.conf
OUTPUT: New configuration file created: new_ns.conf
OUTPUT: New warning file created: warn_ns.conf
WARNINGS: Total number of warnings due to bind commands: 18
WARNINGS: Line numbers which has bind command issues: 305, 306, 706,
    707, 708, 709, 710, 711, 712, 713,
714, 715, 767, 768, 774, 775, 776, 777
root@NS#
<!--NeedCopy-->
```

Conversion warnings

When classic expressions that are included in CLI commands are upgraded to the Advanced policy, the number of characters in the expression might exceed the 1499-character limit. The commands that include expressions longer than 1499 characters fail when the configuration is being applied. You must manually update these commands.

In addition, multiple classic policies can be bound to a given bind point with priority 0 or with equal priority, but the Advanced policy infrastructure does not support a priority value of 0 or policies with the same priority at a given bind point. These commands fail when the configuration is being applied. The commands must be updated manually with the correct priority values.

The line numbers of lines that threw a warning during conversion are listed at the end of the output in a warning line. In addition, a warning file is created in the same directory as the one in which the old and new configuration files reside. The name of the warning file is the same as the name of the input configuration file except that the string warn_ is added as a prefix.

Before you proceed

September 21, 2020

Before configuring expressions and policies, be sure you understand the relevant NetScaler feature and the structure of your data, as follows:

- Read the documentation on the relevant feature.
- Look at the data stream for the type of data that you want to configure.

You may want to run a trace on the type of traffic or content that you want to configure. This will give you an idea of the parameters and values, and operations on these parameters and values, that you need to specify in an expression.

Note: The NetScaler supports either classic or Advanced policy within a feature. You cannot have both types in the same feature. Over the past few releases, some NetScaler features have migrated from using classic policies and expressions to Advanced policy and expressions. If a feature of interest to you has changed to the Advanced policy format, you may have to manually migrate the older information. Following are guidelines for deciding if you need to migrate your policies:

- If you configured classic policies in a version of the Integrated Caching feature prior to release 9.0 and then upgrade to version 9.0 or later, there is no impact. All legacy policies are migrated to the Advanced policy format.
- For other features, you need to manually migrate classic policies and expressions to the Advanced syntax if the feature has migrated to the Advanced policy.

Configure advanced policy infrastructure

September 21, 2020

You can create advanced policies for various NetScaler features, including DNS, Rewrite, Responder, and Integrated Caching, and the clientless access function in the NetScaler Gateway. Policies control the behavior of these features.

When you create a policy, you assign it a name, a rule (an expression), feature-specific attributes, and an action that is taken when data matches the policy. After creating the policy, you determine when it is invoked by binding it globally or to either request-time or response-time processing for a virtual server.

Policies that share the same bind point are known as a *policy bank*. For example, all policies that are bound to a virtual server constitute the policy bank for the virtual server. When binding the policy, you assign it a priority level to specify when it is invoked relative to other policies in the bank. In addition to assigning a priority level, you can configure an arbitrary evaluation order for policies in a bank by specifying Goto expressions.

In addition to policy banks that are associated with a built-in bind point or a virtual server, you can configure *policy labels*. A policy label is a policy bank that is identified by an arbitrary name. You invoke a policy label, and the policies in it, from a global or virtual-server-specific policy bank. A policy label or a virtual-server policy bank can be invoked from multiple policy banks.

For some features, you can use the policy manager to configure and bind policies.

Rules for names in identifiers used in policies

September 21, 2020

The names of identifiers in the named expression, HTTP callout, pattern set, and rate limiting features must begin with an ASCII alphabet or an underscore (*). The remaining characters can be ASCII alphanumeric characters or underscores (*).

The names of these identifiers must not begin with the following reserved words:

- The words ALT, TRUE, or FALSE or the Q or S one-character identifier.
- The special-syntax indicator RE (for regular expressions) or XP (for XPath expressions).
- Expression prefixes, which currently are the following:
  - CLIENT
  - EXTEND
  - HTTP
  - SERVER
  - SYS
  - TARGET

  - TEXT
  - URL
  - MYSQL
  - MSSQL

Additionally, the names of these identifiers cannot be the same as the names of enumeration constants used in the policy infrastructure. For example, the name of an identifier cannot be IGNORECASE, YEAR, or LATIN2_CZECH_CS (a MySQL character set).

Note: The NetScaler appliance performs a case-insensitive comparison of identifiers with these words and enumeration constants. For example, names of the identifiers cannot begin with
TRUE,
True, or
true.

Create or modify a policy

September 21, 2020

All policies have some common elements. Creating a policy consists, at minimum, of naming the policy and configuring a rule. The policy configuration tools for the various features have areas of overlap, but also differences. For the details of configuring a policy for a particular feature, including associating an action with the policy, see the documentation for the feature.

To create a policy, begin by determining the purpose of the policy. For example, you may want to define a policy that identifies HTTP requests for image files, or client requests that contain an SSL certificate. In addition to knowing the type of information that you want the policy to work with, you need to know the format of the data that the policy is analyzing.

Next, determine whether the policy is globally applicable, or if it pertains to a particular virtual server. Also consider the effect that the order in which your policies are evaluated (which will be determined by how you bind the policies) will have on the policy that you are about to configure.

Create a policy by using the CLI

At the command prompt, type the following commands to create a policy and verify the configuration:

```
-  add responder|dns|cs|rewrite|cache policy <policyName> -rule <
   expression> [<feature-specific information>]

-  show rewrite policy <name>
<!--NeedCopy-->
```

Example 1:

```
add rewrite policy "pol_remove-ae" true "act_remove-ae"
Done
> show rewrite policy pol_remove-ae
            Name: pol_remove-ae
            Rule: true
            RewriteAction: act_remove-ae
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
            Bound to: GLOBAL RES_OVERRIDE
            Priority: 90
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

Example 2:

```
add cache policy BranchReportsCachePolicy -rule q{
 http.req.url.query.value("actionoverride").contains("branchReport s")}
  -action cache
Done
show cache policy BranchReportsCachePolicy
            Name: BranchReportsCachePolicy
            Rule: http.req.url.query.value("actionoverride").contains("branchReports")
            CacheAction: CACHE
            Stored in group: DEFAULT
            UndefAction: Use Global
            Hits: 0
            Undef Hits: 0
Done
<!--NeedCopy-->
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, seeConfigure advanced policy expressions: Get started.

Create or modify a policy by using the GUI

1. In the navigation pane, expand the name of the feature for which you want to configure a policy, and then click Policies. For example, you can select Content Switching, Integrated Caching,

DNS, Rewrite, or Responder.

2. In the details pane, click Add, or select an existing policy and click Open. A policy configuration dialog box appears.
3. Specify values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding parameter in "Parameters for creating or modifying a policy.")
4. Click Create, and then click Close.
5. Click Save. A policy is added.

   Note: After you create a policy, you can view the policy's details by clicking the policy entry in the configuration pane. Details that are highlighted and underlined are links to the corresponding entity (for example, a named expression).

Policy configuration examples

September 21, 2020

These examples show how policies and their associated actions are entered at the command line interface. In the GUI, the expressions would appear in the Expression window of the feature-configuration dialog box for the integrated caching or rewrite feature.

Following is an example of creating a caching policy. Note that actions for caching policies are built in, so you do not need to configure them separately from the policy.

```
 add cache policy BranchReportsCachePolicy -rule q{
http.req.url.query.value("actionoverride").contains("branchReports") }
 -action cache
<!--NeedCopy-->
```

Following is an example of a rewrite policy and action:

```
add rewrite action myAction1 INSERT_HTTP_HEADER "myHeader" "
   valueForMyHeader"
add rewrite policy myPolicy1 "http.req.url.contains(\"myURLstring\")"
   myAction1
<!--NeedCopy-->
```

Note: At the command line, quote marks within a policy rule (the expression) must be escaped or delimited with the q delimiter. For more information, see Configure advanced policy expressions: Get started.

Bind policies using advanced policy

July 15, 2021

After defining a policy, the policy can be invoked by binding the policy to a bind point and specifying a priority level. You can bind the policy to only one bind point. A bind point can be global (applying to all configured virtual servers) or bind point can be a specific virtual server (either a load balancing or a content switching virtual server).

The order in which policies are evaluated determines the order in which they are applied. ADC features typically evaluate various policy banks in a particular order. Sometimes, however, other features can affect the order of evaluation. Within a policy bank, the order of evaluation depends on parameter values configured in the policies. Most features apply all the actions associated with a policy if the evaluated result matches with the data being processed.

> **Note:**
>
> Integrated caching feature is an exception.

Feature-specific differences in policy bindings

You can bind policies to a built-in, global bind points (or banks), specific virtual servers, or policy labels.

However, the Citrix ADC features differ in the type of bindings that are available. The following table summarizes policy bindings in Citrix ADC features that use policies.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
DNS	none	DNS policies	Global	To determine how to perform DNS resolution for requests.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
Content Switching (Note: This feature can support either or classic or Advanced policies, but not both.)	Content Switching (CS)	Content Switching policies	Content switching or cache redirection virtual server; Policy label	To determine what server or group of servers is responsible for serving responses, based on characteristics of an incoming request. Request characteristics include device type, language, cookies, HTTP method, content type, and associated cache server.
Integrated Caching	none	Caching policies	Global override, Global default, Policy label, load balancing, content switching, or SSL offload virtual server	To determine whether HTTP responses can be stored in, and served from, the Citrix ADC appliance's integrated cache.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
Responder	none	Responder policies	Global override, Global default, Policy label, load balancing, content switching, or SSL offload virtual server	To configure the behavior of the Responder function.
Rewrite	none	Rewrite policies	Global override, Global default, Policy label, load balancing, content switching, or SSL offload virtual server	To identify HTTP data that you want to modify before serving. The policies provide rules for modifying the data. For example, you can modify HTTP data to redirect a request to a selected server based on the address of the incoming request, or to mask server information in a response for security purposes.

Feature Name	Virtual Servers Configured in the Feature	Policies Configured in the Feature	Bind Points Configured for the Policies	Use of Policies in the Feature
URL Transform function in the Rewrite feature	none	Transformation policies	Global override, Global default, Policy label	To identify URLs in HTTP transactions and text files for evaluating whether a URL should be altered.
Citrix Gateway (clientless VPN functions only)	VPN server	Clientless Access policies	VPN Global, VPN server	To determine how the Citrix Gateway performs authentication, authorization, auditing, and other functions, and to define rewrite rules for general Web access using the Citrix Gateway.

Bind points and order of evaluation

For a policy to take effect, you must ensure that the policy is invoked at some point during processing. To do so, you associate the policy with a bind point. The collection of policies that is bound to a bind point is known as a policy bank.

Following are the bind points that the Citrix ADC evaluates, listed in the typical order of evaluation within a policy bank

1. **Request-time override.** When a request flows through a feature, the Citrix ADC first evaluates request-time override policies for the feature.
2. **Request-time Load Balancing virtual server.** If policy evaluation cannot be completed after all the request-time override policies have been evaluated, the Citrix ADC processes request-time policies for load balancing virtual servers.

3. **Request-time Content Switching virtual server.** If policy evaluation cannot be completed after all the request-time policies for load balancing virtual servers have been evaluated, the Citrix ADC processes request-time policies for content switching virtual servers.
4. **Request-time default.** If policy evaluation cannot be completed after all request-time, virtual server-specific policies have been evaluated, the Citrix ADC processes request-time Advanced policies.
5. **Response-time override.** At response time, the Citrix ADC starts with policies that are bound to the response-time override bind point.
6. **Response-time Load Balancing virtual server.** If policy evaluation cannot be completed after all response-time override policies have been evaluated, the Citrix ADC process the response-time policies for load balancing virtual servers.
7. **Response-time Content Switching virtual server.** If policy evaluation cannot be completed after all policies have been evaluated for load balancing virtual servers, the Citrix ADC process the response-time policies for content switching virtual servers.
8. **Response-time default.** If policy evaluation cannot be completed after all response-time, virtual-server-specific policies have been evaluated, the Citrix ADC processes response-time Advanced policies.

Policy evaluation across features

In addition to evaluate policies within a feature, if you bind policies to a content switching virtual server, it is important that these policies are evaluated before other policies. Binding a policy to a content switching virtual server produces a different result in Citrix ADC versions 9.0.x and later than in 8.x versions. In Citrix ADC 9.0 and later versions, evaluation occurs as follows:

- Content switching policies are evaluated before other policies. If a content switching policy evaluates to TRUE, the target load balancing virtual server is selected.
- If all content switching policies evaluate to FALSE, the default load balancing virtual server under the content switching VIP is selected.

After a target load balancing virtual server is selected by the content switching process, policies are evaluated in the following order:

1. Policies that are bound to the global override bind point.
2. Policies that are bound to the default load balancing virtual server.
3. Policies that are bound to the target content switching virtual server.
4. Policies that are bound to the global default bind point.

To be sure that the policies are evaluated in the intended order, follow these guidelines:

- Make sure that the default load balancing virtual server is not directly reachable from the outside; for example, the virtual server IP address can be 0.0.0.0.

- To prevent exposing internal data on the load balancing default virtual server, configure a policy to respond with a "503 Service Unavailable" status and bind it to the default load balancing virtual server.

Entries in a policy bank

Each entry in a policy bank has, at minimum, a policy and a priority level. You can also configure entries that change the priority-based evaluation order, and you can configure entries that invoke external policy banks.

The following table summarizes each entry in a policy bank.

Policy Name	Priority	Goto Expression	Invocation Type	Policy Bank to Be Invoked
The policy name, or a "dummy" policy named NOPOLICY. The **NOPOLICY** entry controls evaluation flow without processing a rule.	An integer.	Optional. Identifies the next policy in the bank to evaluate, or ends any further evaluation	Optional. Indicates that an external policy bank will be invoked. This field restricts the choices to a global policy label or a virtual server.	Optional. Used with Invocation Type. This is the label for a policy bank or a virtual server name. The Citrix ADC returns to the current bank after processing the external bank.

If the policy evaluates to TRUE, the Citrix ADC stores the action that is associated with the policy. If the policy evaluates to FALSE, the Citrix ADC evaluates the next policy. If the policy is neither TRUE nor FALSE, the Citrix ADC uses the associated Under (undefined) action.

Evaluation order within a policy bank

Within a policy bank, the evaluation order depends on the following parameters:

- **A priority.**

  The most minimal amount of information about evaluation order is a numeric priority level. The lower the number, the higher the priority.

- **A Goto expression.**

If supplied, the Goto expression indicates the next policy to be evaluated, typically within the same policy bank. Goto expressions can only proceed forward in a bank. To prevent looping, a policy bank configuration is not valid if a Goto statement point backwards in the bank.

- **Invocation of other policy banks.**

  Any entry can invoke an external policy bank. The Citrix ADC provides a built-in entity named NOPOLICY that does not have a rule. You can add a NOPOLICY entry in a policy bank when you want to invoke another policy bank, but do not want to process any other rules prior to the invocation. You can have multiple NOPOLICY entries in multiple policy banks.

Values for a Goto expression are as follows:

- **NEXT.**

  This keyword selects the policy with the next higher priority level in the current policy bank. The Policies are evaluated in priority order from lower numbered priority to higher numbered priority.

- **An integer.**

  If you supply an integer, it must match the priority level of another policy in the current policy bank.

- **END.**

  This keyword stops evaluation after processing the current policy, and no additional policies in this bank are processed.

- **Blank.**

  If the Goto expression is empty, it is the same as specifying END.

- **A numeric expression.**

  This is an advanced policy expression that resolves to a priority number for another policy in the current bank.

- **USE_INVOCATION_RESULT.**

  This phrase can be used only if you are invoking an external policy bank. Entering this phrase causes the Citrix ADC to perform one of the following actions:

  - If the final Goto in the invoked policy bank has a value of END or is empty, the invocation result is END, and evaluation stops.
  - If the final Goto expression in the invoked policy bank is anything other than END, the Citrix ADC performs a NEXT.

The following table illustrates a policy bank that uses Goto statements and policy bank invocations.

Polic Name	Priori	Goto	Invoc	Polic Bank to Be Invoke	—	——–	—-	—-	—-					
ClientCertificatePolicy (rule: does the request contain a client certificate?)	100	300	None	None										
Subn (rule: is the clien from a private subnet?)	200	NEXT	None	None	NOPO	300	USE INVOCATION RESULT	Requ virtual serve	My_R	NOPO	350	USE INVOCATION RESULT	Polic Label	My_Policy_Label
WorkingHoursPolicy (rule: is it working hours?)	400	END	None	None										

Table 3. Example of a Policy Bank That Uses Gotos and External Bank Invocations

How policy evaluation ends

Evaluation of a policy bank ends when the Citrix ADC appliance executes one of the following steps:

- If a policy evaluation is "TRUE," it invokes an external policy bank and its Goto statement value is "USE_INVOCATION_RESULT," and if another policy in the external policy bank also evaluates "TRUE" and Goto statement value is "END," then after returning from the external policy bank, no further policy will be evaluated.
- An external policy bank is invoked, its evaluation returns an END, and the Goto statement uses a value of USE_INVOCATION_RESULT or END. Evaluation continues with the next policy bank for this feature. For example, if the current bank is the request-time override bank, the Citrix ADC next evaluates request-time policy banks for the virtual servers.
- The Citrix ADC has walked through all the policy banks in this feature, but has not encountered an END.

If this is the last entry to be evaluated in this policy bank, the Citrix ADC proceeds to the next feature.

How features use actions after policy evaluation

After evaluating all relevant policies for a particular data point (for example, an HTTP request), the Citrix ADC stores all the actions that are associated with any policy that matched the data.

For most features, all the actions from matching policies are applied to a traffic packet as it leaves the Citrix ADC. The Integrated Caching feature only applies one action: CACHE or NOCACHE. This action is associated with the policy with the lowest priority value in the "highest priority" policy bank (for example, request-time override policies are applied before virtual server-specific policies).

Bind a policy globally

September 21, 2020

The following binding procedures are typical. However, refer to the documentation for the feature of interest to you for complete instructions.

Bind an integrated caching policy globally by using the CLI

At the command prompt, type the following commands to bind an Integrated Caching policy and verify the configuration:

```
-  bind cache global <policy> -priority <positiveInteger> [-type
   REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]
```

```

-  show cache global
<!--NeedCopy-->
```

Example:

```
> bind cache global _nonPostReq -priority 100 -type req_default
     Done
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

The type argument is optional to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

Bind a rewrite policy globally by using the CLI

At the command prompt, type the following commands to bind a Rewrite policy and verify the configuration:

```
-  bind rewrite global <policyName> <priority> [-type REQ_OVERRIDE |
   REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]

-  show rewrite global
<!--NeedCopy-->
```

Example:

```
bind rewrite global pol_remove-pdf 100
Done
> show rewrite global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1

Done
```

```
<!--NeedCopy-->
```

The type argument is optional for globally bound policies, to maintain backward compatibility. If you omit the type, the policy is bound to REQ_DEFAULT or RES_DEFAULT, depending on whether the policy rule is a response-time or a request-time expression.

Bind a compression policy globally by using the CLI

At the command prompt, type the following commands to bind a compression policy and verify the configuration:

```
-  bind cmp global <policyName> -priority <positiveInteger> [-type
    REQ_OVERRIDE | REQ_DEFAULT | RES_OVERRIDE | RES_DEFAULT]

-  show cmp global
<!--NeedCopy-->
```

Example:

```
> bind cmp global cmp_pol_1 -priority 100
Done
> show cmp policy cmp_pol_1
            Name: cmp_pol_1
            Rule: HTTP.REQ.URL.SUFFIX.EQ("BMP")
            Response Action: COMPRESS
            Hits: 0

            Policy is bound to following entities
            1) GLOBAL REQ_DEFAULT
            Priority: 100
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

Bind a responder policy globally by using the CLI

At the command prompt, type the following commands to bind a Responder policy and verify the configuration:

```
-  bind responder global <policyName> <priority> [-type OVERRIDE |
    DEFAULT ]

-  show responder global
```

```
<!--NeedCopy-->
```

Example:

```
bind responder global pol404Error1 200
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Bind a DNS policy globally by using the CLI

At the command prompt, type the following commands to bind a DNS policy and verify the configuration:

```
-  bind dns global <policyName> <priority>

-  show dns global
<!--NeedCopy-->
```

Example:

```
> bind dns global pol_ddos_drop1 150
Done
> show dns global
    Policy name : pol_ddos_drop
            Priority : 100
            Goto expression : END
    Policy name : pol_ddos_drop1
            Priority : 150
Done
<!--NeedCopy-->
```

Bind an integrated caching, responder, rewrite, or compression policy globally by using the GUI

1. In the navigation pane, click the name of the feature for which you want to bind the policy.
2. In the details pane, click <Feature Name> policy manager.
3. In the Policy Manager dialog box, select the bind point to which you want to bind the policy (for example, for Integrated Caching, Rewrite, or Compression, you could select Request and

Default Global). The Responder does not differentiate between request-time and response-time policies.

4. Click Insert Policy and, from the Policy Name pop-up menu, select the policy name. A priority is assigned automatically to the policy, but you can click the cell in the Priority column and drag it anywhere within the dialog box if you want the policy to be evaluated after other policies in this bank. The priority is automatically reset. Note that priority values within a policy bank must be unique.
5. Click Apply Changes.
6. Click Close. A message in the status bar indicates that the policy is bound successfully.

Bind a DNS policy globally by using the GUI

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the global bindings dialog box, click Insert Policy, and select the policy that you want to bind globally.
4. Click in the Priority field and enter the priority level.
5. Click OK. A message in the status bar indicates that the policy is bound successfully.

Bind a policy to a virtual server

September 21, 2020

A globally bound policy applies to all load balancing and content switching virtual servers.

Note that when binding a policy to a virtual server, you must identify it as a request-time or a response-time policy.

Bind a policy to a load balancing or content switching virtual server by using the CLI

At the command prompt, type the following commands to bind a policy to a load balancing or content switching virtual server and verify the configuration:

```
-  bind lb|cs vserver <virtualServerName> -policyName <policyName> -
    priority <positiveInteger> -type REQUEST|RESPONSE

-  show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
Done
> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
            Time since last state change: 28 days, 01:57:26.350
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      Policy : ns_cmp_msapp Priority:50
    2)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

Bind a policy to an SSL offload virtual server by using the CLI

At the command prompt, type the following commands to bind a policy to an SSL offload virtual server and verify the configuration:

```
bind ssl vserver <vServerName>@ -policyName <policyName> -priority <positiveInteger
>
```

Bind a policy to a virtual server by using the GUI

1. In the navigation pane, expand Traffic Management > Load Balancing, Traffic Management > Content Switching, Traffic Management > SSL Offload, Security > AAA- Application Traffic, or NetScaler Gateway, and then click Virtual Servers.
2. In the details pane, double-click the virtual server to which you want to bind the policy, and then click Open.

3. On the Policies tab, click the icon for the type of policy that you want to bind (the choices are feature-specific), and then click the name of the policy. Note that for some features, you can bind both classic policies and policies that use the default syntax to the virtual server.
4. If you are binding a policy to a Content Switching virtual server, in the Target field select a load balancing virtual server to which traffic that matches the policy is sent.
5. Click OK. A message in the status bar indicates that the policy is bound successfully.

Display policy bindings

September 21, 2020

You can display policy bindings to verify that they are correct.

Display policy bindings by using the CLI

At the command prompt, type the following commands to display policy bindings and verify the configuration:

`show rewrite policy <name>`

Example:

```
> show rewrite policy pol_remove-pdf
        Name: pol_remove-pdf
        Rule: http.req.url.contains(".pdf")
        RewriteAction: act_remove-ae
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0
        Bound to: GLOBAL REQ_DEFAULT
        Priority: 100
        GotoPriorityExpression: END
 Done
>
<!--NeedCopy-->
```

Display global policy bindings for integrated caching, rewrite, or responder by using the GUI

1. In the navigation pane, expand the feature that contains the policy that you want to view, and then click Policies.

2. In the details pane, click the policy. Bound policies have a check mark next to them.
3. At the bottom of the page, under Details, next to Bound to, view the entity to which the policy is bound.

Display global policy bindings for DNS or clientless access in the NetScaler Gateway by using the GUI

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.

Display global policy bindings for content switching by using the GUI

1. Navigate to Traffic Management > Content Switching > Policies.
2. In detailed pane, select policy.
3. In the details pane, click Show Bindings.

Unbind a policy

September 21, 2020

If you want to re-assign a policy or delete it, you must first remove its binding.

Unbind an integrated caching, rewrite, or compression advanced policy globally by using the CLI

At the command prompt, type the following commands to unbind an integrated caching, rewrite, or compression Advanced policy globally and verify the configuration:

```
-  unbind cache|rewrite|cmp global <policyName> [-type req_override|
    req_default|res_override|res_default] [-priority <positiveInteger>]

-  show cache|rewrite|cmp global
<!--NeedCopy-->
```

Example:

```
> unbind cache global_nonPostReq
Done
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

The priority is required only for the "dummy" policy named NOPOLICY.

Unbind a responder policy globally by using the CLI

At the command prompt, type the following commands to unbind a responder policy globally and verify the configuration:

```
-  unbind responder global <policyName> [-type override|default] [-
   priority <positiveInteger>]

-  show responder global
<!--NeedCopy-->
```

Example:

```
> unbind responder global pol404Error
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

The priority is required only for the "dummy" policy named NOPOLICY.

Unbind a DNS policy globally by using the CLI

At the command prompt, type the following commands to unbind a DNS policy globally and verify the configuration:

```
-  unbind responder global <policyName>

-  unbind responder global
<!--NeedCopy-->
```

Example:

```
unbind dns global dfgdfg
Done
show dns global
    Policy name : dfgdfggfhg
            Priority : 100
            Goto expression : END
Done
<!--NeedCopy-->
```

Unbind an advanced policy from a virtual server by using the CLI

At the command prompt, type the following commands to unbind an Advanced policy from a virtual server and verify the configuration:

```
-  unbind cs vserver <name> -policyName <policyName> [-priority <
    positiveInteger>] [-type REQUEST|RESPONSE]

-  show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
unbind cs vserver vs-cont-switch  -policyName pol1
Done
> show cs vserver vs-cont-switch
            vs-cont-switch (10.102.29.10:80) - HTTP Type: CONTENT
            State: UP
            Last state change was at Wed Aug 19 08:56:55 2009 (+18 ms)
            Time since last state change: 0 days, 02:47:55.750
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            State Update: DISABLED
            Default:        Content Precedence: RULE
            Vserver IP and Port insertion: OFF
            Case Sensitivity: ON
            Push: DISABLED  Push VServer:
            Push Label Rule: none
Done
<!--NeedCopy-->
```

The priority is required only for the “dummy” policy named NOPOLICY.

Unbind an integrated caching, responder, rewrite, or compression Advanced policy globally by using the GUI

1. In the navigation pane, click the feature with the policy that you want to unbind (for example, Integrated Caching).
2. In the details pane, click <Feature Name> policy manager.
3. In the **Policy Manager** dialog box, select the bind point with the policy that you want to unbind, for example, Advanced Global.
4. Click the policy name that you want to unbind, and then click Unbind Policy.
5. Click **Apply Changes**.
6. Click **Close**. A message in the status bar indicates that the policy is unbound successfully.

Unbind a DNS policy globally by using the GUI

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the Global Bindings dialog box, select policy and click unbind policy.
4. Click OK. A message in the status bar indicates that the policy is unbinded successfully.

Unbind an advanced policy from a load balancing or content switching virtual server by using the GUI

1. Navigate to **Traffic Management,** and expand Load Balancing or Content Switching, and then click **Virtual Servers**.
2. In the details pane, double-click the virtual server from which you want to unbind the policy.
3. On the **Policies** tab, in the **Active** column, clear the check box next to the policy that you want to unbind.
4. Click **OK**. A message in the status bar indicates that the policy is unbinded successfully.

Create policy labels

September 21, 2020

In addition to the built-in bind points where you set up policy banks, you can also configure user-defined policy labels and associate policies with them.

Within a policy label, you bind policies and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. The NetScaler also permits you to define an arbitrary evaluation order as follows:

- You can use "goto" expressions to point to the next entry in the bank to be evaluated after the current one.
- You can use an entry in a policy bank to invoke another bank.

Each feature determines the type of policy that you can bind to a policy label, the type of load balancing virtual server that you can bind the label to, and the type of content switching virtual server from which the label can be invoked. For example, a TCP policy label can only be bound to a TCP load balancing virtual server. You cannot bind HTTP policies to a policy label of this type. And you can invoke a TCP policy label only from a TCP content switching virtual server.

After configuring a new policy label, you can invoke it from one or more banks for the built-in bind points.

Create a caching policy label by using the CLI

At the command prompt, type the following commands to create a Caching policy label and verify the configuration:

```
-  add cache policylabel <labelName> -evaluates req|res

-  show cache policylabel<labelName>
<!--NeedCopy-->
```

Example:

```
> add cache policylabel lbl-cache-pol -evaluates req
Done

> show cache policylabel lbl-cache-pol
            Label Name: lbl-cache-pol
            Evaluates: REQ
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Create a content switching policy label by using the CLI

At the command prompt, type the following commands to create a Content Switching policy label and verify the configuration:

```
-  add cs policylabel <labelName> http|tcp|rtsp|ssl

-  show cs policylabel <labelName>
```

```
<!--NeedCopy-->
```

Example:

```
> add cs policylabel lbl-cs-pol http
Done
> show cs policylabel lbl-cs-pol
            Label Name: lbl-cs-pol
            Label Type: HTTP
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Create a rewrite policy label by using the CLI

At the command prompt, type the following commands to create a Rewrite policy label and verify the configuration:

```
-  add rewrite policylabel <labelName> http_req|http_res|url|text|clientless_vpn_req|clientless_vpn_res

-  show rewrite policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
> add rewrite policylabel lbl-rewrt-pol http_req
Done

> show rewrite policylabel lbl-rewrt-pol
            Label Name: lbl-rewrt-pol
            Transform Name: http_req
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Create a responder policy label by using the CLI

At the command prompt, type the following commands to create a Responder policy label and verify the configuration:

```
-  add responder policylabel <labelName>

-  show responder policylabel <labelName>
<!--NeedCopy-->
```

Example:

```
> add responder policylabel lbl-respndr-pol
Done

> show responder policylabel lbl-respndr-pol
            Label Name: lbl-respndr-pol
            Number of bound policies: 0
            Number of times invoked: 0
Done
<!--NeedCopy-->
```

Note: Invoke this policy label from a policy bank. For more information, see the "Binding a Policy to a Policy Label" section.

Create a policy label by using the GUI

1. In the navigation pane, expand the feature for which you want to create a policy label, and then click Policy Labels. The choices are Integrated Caching, Rewrite, Content Switching, or Responder.
2. In the details pane, click Add.
3. In the Name box, enter a unique name for this policy label.
4. Enter feature-specific information for the policy label. For example, for Integrated Caching, in the Evaluates drop-down menu, you would select REQ if you want this policy label to contain request-time policies, or select RES if you want this policy label to contain response-time policies. For Rewrite, you would select a Transform name.
5. Click Create.
6. Configure one of the built-in policy banks to invoke this policy label. For more information, see the "Binding a Policy to a Policy Label" section. A message in the status bar indicates that the policy label is created successfully.

Bind a policy to a policy label

As with policy banks that are bound to the built-in bind points, each entry in a policy label is a policy that is bound to the policy label. As with policies that are bound globally or to a vserver, each policy that is bound to the policy label can also invoke a policy bank or a policy label that is evaluated after the current entry has been processed. The following table summarizes the entries in a policy label.

- **Name**. The name of a policy, or, to invoke another policy bank without evaluating a policy, the "dummy" policy name NOPOLICY.

  You can specify NOPOLICY more than once in a policy bank, but you can specify a named policy only once.

- **Priority**. An integer. This setting can work with the Goto expression.
- **Goto Expression**. Determines the next policy to evaluate in this bank. You can provide one of the following values:

  - **NEXT**. Go to the policy with the next higher priority.
  - **END**. Stop evaluation.
  - **USE_INVOCATION_RESULT**. Applicable if this entry invokes another policy bank. If the final Goto in the invoked bank has a value of END, evaluation stops. If the final Goto is anything other than END, the current policy bank performs a NEXT.
  - **Positive number:** The priority number of the next policy to be evaluated.
  - **Numeric expression**. An expression that produces the priority number of the next policy to be evaluated.

  The Goto can only proceed forward in a policy bank.

  If you omit the Goto expression, it is the same as specifying END.

- **Invocation Type**. Designates a policy bank type. The value can be one of the following:

  - **Request Vserver**. Invokes request-time policies that are associated with a virtual server.
  - **Response Vserver**. Invokes response-time policies that are associated with a virtual server.
  - **Policy label**. Invokes another policy bank, as identified by the policy label for the bank.

- **Invocation Name**. The name of a virtual server or a policy label, depending on the value that you specified for the Invocation Type.

Configure a policy label or virtual server policy bank

July 7, 2021

After you have created policies, and created policy banks by binding the policies, you can perform additional configuration of policies within a label or policy bank. For example, before you configure invocation of an external policy bank, you might want to wait until you have configured that policy bank.

This topic includes the following sections:

- Configure a policy label
- Configure a policy bank for a virtual server

Configure a policy label

A policy label consists of a set of policies and invocations of other policy labels and virtual server-specific policy banks. An Invoke parameter enables you to invoke a policy label or a virtual server-specific policy bank from any other policy bank. A special-purpose NoPolicy entry enables you to invoke an external bank without processing an expression (a rule). The NoPolicy entry is a "dummy" policy that does not contain a rule.

For configuring policy labels from the NetScaler command line, note the following elaborations of the command syntax:

- gotoPriorityExpression is configured as described in Table 2. Format of Each Entry in a Policy Bank of the section "Entries in a Policy Bank" in Bind policies using advanced policy.
- The type argument is required. This is unlike binding a conventional policy, where this argument is optional.
- You can invoke the bank of policies that are bound to a virtual server by using the same method as you use for invoking a policy label.

Configure a policy label by using the CLI

At the command prompt, type the following commands to configure a policy label and verify the configuration:

```
-  bind cache|rewrite|responder policylabel <policylabelName> -policyName <policyName> -priority <priority> [-gotoPriorityExpression <gotopriorityExpression>] [-invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>]

-  show cache|rewrite|responder policylabel <policylabelName>
<!--NeedCopy-->
```

Example:

```
bind cache policylabel _reqBuiltinDefaults -policyName  _nonGetReq -priority 100
Done
show cache policylabel _reqBuiltinDefaults
            Label Name: _reqBuiltinDefaults
            Evaluates: REQ
            Number of bound policies: 3
            Number of times invoked: 0
    1)      Policy Name: _nonGetReq
            Priority: 100
            GotoPriorityExpression: END
    2)      Policy Name: _advancedConditionalReq
            Priority: 200
            GotoPriorityExpression: END

    3)      Policy Name: _personalizedReq
            Priority: 300
            GotoPriorityExpression: END
Done
<!--NeedCopy-->
```

Invoke a policy label from a rewrite policy bank with a NOPOLICY entry by using the CLI

At the command prompt, type the following commands to invoke a policy label from a Rewrite policy bank with a NOPOLICY entry and verify the configuration:

```
-  bind rewrite global <policyName> <priority> <gotoPriorityExpression> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>

-  show rewrite global
<!--NeedCopy-->
```

Example:

```
> bind rewrite global NOPOLICY 100 -type REQ_DEFAULT -invoke policylabel lbl-rewrt-pol
Done
> show rewrite global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

    2)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1
Done
<!--NeedCopy-->
```

Invoke a policy label from an Integrated Caching policy bank by using the CLI

At the command prompt, type the following commands to invoke a policy label from an Integrated Caching policy bank and verify the configuration:

```
-  bind cache global NOPOLICY -priority <priority> -gotoPriorityExpression <gotopriorityExpression> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT -invoke reqvserver|resvserver|policylabel <policyLabelName>|<vserverName>

-  show cache global
<!--NeedCopy-->
```

Example:

```
bind cache global NOPOLICY -priority 100 -gotoPriorityExpression END -type REQ_DEFAULT -invoke policylabel lbl-cache-pol
Done
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Invoke a policy label from a Responder policy bank by using the CLI

At the command prompt, type the following commands to invoke a policy label from a Responder policy bank and verify the configuration:

```
-  bind responder global NOPOLICY <priority> <gotopriorityExpression> -type OVERRIDE|DEFAULT -invoke vserver|policylabel <policyLabelName>|<vserverName>

-  show responder global
<!--NeedCopy-->
```

Example:

```
> bind responder global NOPOLICY 100 NEXT -type DEFAULT -invoke policylabel lbl-respndr-pol
Done
```

```
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
<!--NeedCopy-->
```

Configure a policy label by using the GUI

1. In the navigation pane, expand the feature for which you want to configure a policy label, and then click Policy Labels. The choices are Integrated Caching, Rewrite, or Responder.
2. In the details pane, double-click the label that you want to configure.
3. If you are adding a new policy to this policy label, click Insert Policy, and in the Policy Name field, select New Policy. For more information about adding a policy, see Create or modify a policy. Note that if you are invoking a policy bank, and do not want a rule to be evaluated prior to the invocation, click Insert Policy, and in the Policy Name field select NOPOLICY.
4. For each entry in this policy label, configure the following:
   - **Policy Name:**

     This is already determined by the Policy Name, new policy, or NOPOLICY entry that you inserted in this bank.
   - **Priority:**

     A numeric value that determines either an absolute order of evaluation within the bank, or is used in conjunction with a Goto expression.
   - **Expression:**

     The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see
     Configure advanced policy expressions: Get started.
   - **Action:**

     The action to be taken if this policy evaluates to TRUE.
   - **Goto Expression:**

     Optional. Used to augment the Priority level to determine the next policy or policy bank to evaluate. For more information on possible values for a Goto expression, see Table 2. Format of Each Entry in a Policy Bank of the section "Entries in a Policy Bank" in
     Bind policies using advanced policy.

- **Invoke:**

  Optional. Invokes another policy bank.

5. Click **OK**. A message in the status bar indicates that the policy label is configured successfully.

Configure a policy bank for a virtual server

You can configure a bank of policies for a virtual server. The policy bank can contain individual policies, and each entry in the policy bank can optionally invoke a policy label or a bank of policies that you configured for another virtual server. If you invoke a policy label or policy bank, you can do so without triggering an expression (a rule) by selecting a NOPOLICY "dummy" entry instead of a policy name.

Add policies to a virtual server policy bank by using the CLI

At the command prompt, type the following commands to add policies to a virtual server policy bank and verify the configuration:

```
-  bind lb|cs vserver <virtualServerName> <serviceType> [-policyName <
   policyName>] [-priority <positiveInteger>] [-gotoPriorityExpression
   <expression>] [-type REQUEST|RESPONSE]

-  show lb|cs vserver <virtualServerName>
<!--NeedCopy-->
```

Example:

```
add lb vserver vs-cont-sw TCP
Done
show lb vserver vs-cont-sw
            vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
            State: DOWN
            Last state change was at Wed Aug 19 10:04:02 2009 (+279 ms)
            Time since last state change: 0 days, 00:02:14.420
            Effective State: DOWN
            Client Idle Timeout: 9000 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Connection Failover: DISABLED
Done
```

```
<!--NeedCopy-->
```

Invoke a policy label from a virtual server policy bank with a NOPOLICY entry by using the CLI

At the command prompt, type the following commands to invoke a policy label from a virtual server policy bank with a NOPOLICY entry and verify the configuration:

```
-  bind lb|cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|
   NOPOLICY-CACHE|NOPOLICY-RESPONDER -priority <integer> -type REQUEST|
   RESPONSE -gotoPriorityExpression <gotopriorityExpression> -invoke
   reqVserver|resVserver|policyLabel <vserverName>|<labelName>

-  show lb vserver
<!--NeedCopy-->
```

Example:

```
> bind lb vserver vs-cont-sw -policyname NOPOLICY-REWRITE -priority 200
    -type REQUEST -gotoPriorityExpression NEXT -invoke policyLabel lbl-
   rewrt-pol
Done
<!--NeedCopy-->
```

Configure a virtual server policy bank by using the GUI

1. In the left navigation pane, expand ** **Traffic Management** > **Load Balancing, Traffic Management** > **Content Switching, Traffic Management** > **SSL Offload, Security** > **AAA - Application Traffic**, or **NetScaler Gateway**, as appropriate, and then click **Virtual Servers.**
2. In the details pane, select the virtual server that you want to configure, and then click **Open**.
3. In the **Configure Virtual Server** dialog box click the **Policies** tab.
4. To create a new policy in this bank, click the icon for the type of policy or policy label that you want to add to the virtual server's bank of policies, click **Insert Policy**. Note that if you want to invoke a policy label without evaluating a policy rule, select the NOPOLICY "dummy" policy.
5. To configure an existing entry in this policy bank, enter the following:
   - **Priority:**
     A numeric value that determines either an absolute order of evaluation within the bank or is used in conjunction with a Goto expression.
   - **Expression:**
     The policy rule. Policy expressions are described in detail in the following chapters. For an introduction, see
     Configuring Advanced Policy Expressions: Getting Started.

- **Action:**
  The action to be taken if this policy evaluates to TRUE.
- **Goto Expression:**
  Optional. Determines the next policy or policy bank evaluate. For more information on possible values for a Goto expression, see the section "Entries in a Policy Bank" in Bind policies using advanced policy.
- **Invoke:**
  Optional. To invoke another policy bank, select the name of the policy label or virtual server policy bank that you want to invoke.

6. Click **OK**. A message in the status bar indicates that the policy is configured successfully.

Invoke or remove a policy label or virtual server policy bank

September 21, 2020

Unlike a policy, which can only be bound once, you can use a policy label or a virtual server's policy bank any number of times by invoking it. Invocation can be performed from two places:

- From the binding for a named policy in a policy bank.
- From the binding for a NOPOLICY "dummy" entry in a policy bank.

Typically, the policy label must be of the same type as the policy from which it is invoked. For example, you would invoke a responder policy label from a responder policy.

Note: When binding or unbinding a global NOPOLICY entry in a policy bank at the command line, you specify a priority to distinguish one NOPOLICY entry from another.

Invoke a rewrite or integrated caching policy label by using the CLI

At the command prompt, type the one of the following commands to invoke a rewrite or integrated caching policy label and verify the configuration:

```
-  bind cache global <policy> -priority <positive_integer> [-gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel <label_name>

-  bind rewrite global<policy> -priority <positive_integer> [-gotoPriorityExpression <expression>] -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT] -invoke reqvserver|resvserver|policylabel <label_name>

-  show cache global|show rewrite global
<!--NeedCopy-->
```

Example:

```
> bind cache global _nonPostReq2 -priority 100 -type req_override -invoke
    policylabel lbl-cache-pol
Done
> show cache global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

    2)      Global bindpoint: RES_DEFAULT
            Number of bound policies: 1

    3)      Global bindpoint: REQ_OVERRIDE
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Invoke a responder policy label by using the CLI

At the command prompt, type the following commands to invoke a responder policy label and verify the configuration:

```
-  bind responder global <policy_Name> <priority_as_positive_integer> [<gotoPriorityExpression>] -type REQ_OVERRIDE|REQ_DEFAULT|OVERRIDE|DEFAULT -invoke vserver|policylabel <label_name>

-  show responder global
<!--NeedCopy-->
```

Example:

```
> bind responder global pol404Error1 300 -invoke policylabel lbl-respndr-pol
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 2

Done
```

```
<!--NeedCopy-->
```

Invoke a virtual server policy bank by using the CLI

At the command prompt, type the following commands to invoke a Virtual Server Policy Bank and verify the configuration:

```
-  bind lb vserver <vserver_name> -policyName <policy_Name> -priority <positive_integer> [-gotoPriorityExpression <expression>] -type REQUEST|RESPONSE -invoke reqvserver|resvserver|policylabel <policy_Label_Name>

-  bind lb vserver <vserver_name>
<!--NeedCopy-->
```

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 100
Done

> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+166 ms)
            Time since last state change: 28 days, 06:37:49.250
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)       0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

    2)      Policy : pol-ssl Priority:0
    3)      Policy : ns_cmp_msapp Priority:100
    4)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

Remove a rewrite or integrated caching policy label by using the CLI

At the command prompt, type one of the following commands to remove a rewrite or integrated caching policy label and verify the configuration:

```
-  unbind rewrite global <policyName> -priority <positiveInteger> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  unbind cache global <policyName> -priority <positiveInteger> -type REQ_OVERRIDE|REQ_DEFAULT|RES_OVERRIDE|RES_DEFAULT

-  show rewrite global|show cache global
<!--NeedCopy-->
```

Example:

```
> unbind rewrite global NOPOLICY -priority 100 -type REQ_OVERRIDE
> show rewrite global
Done
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Remove a responder policy label by using the CLI

At the command prompt, type the following commands to remove a responder policy label and verify the configuration:

```
-  unbind responder global <policyName> -priority <positiveInteger> -type OVERRIDE|DEFAULT

-  show responder global
<!--NeedCopy-->
```

Example:

```
> unbind responder global NOPOLICY -priority 100 -type REQ_DEFAULT
```

```
Done
> show responder global
    1)      Global bindpoint: REQ_DEFAULT
            Number of bound policies: 1

Done
<!--NeedCopy-->
```

Remove a virtual server policy label by using the CLI

At the command prompt, type one of the following commands to remove a Virtual Server policy label and verify the configuration:

```
-  unbind lb vserver <virtualServerName> -policyName NOPOLICY-REWRITE|NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <positiveInteger>

-  unbind cs vserver <virtualServerName> -policyName NOPOLICY-REWRITE|NOPOLICY-RESPONDER|NOPOLICY-CACHE -type REQUEST|RESPONSE -priority <positiveInteger>

-  show lb vserver|show cs vserver
<!--NeedCopy-->
```

Example:

```
> unbind lb vserver lbvip -policyName ns_cmp_msapp -priority 200
Done
> show lb vserver lbvip
            lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
            State: DOWN
            Last state change was at Wed Jul 15 05:54:24 2009 (+161 ms)
            Time since last state change: 28 days, 06:47:54.600
            Effective State: DOWN
            Client Idle Timeout: 180 sec
            Down state flush: ENABLED
            Disable Primary Vserver On Down : DISABLED
            Port Rewrite : DISABLED
            No. of Bound Services :  0 (Total)        0 (Active)
            Configured Method: LEASTCONNECTION
            Mode: IP
            Persistence: NONE
            Vserver IP and Port insertion: OFF
            Push: DISABLED  Push VServer:
```

```
            Push Multi Clients: NO
            Push Label Rule: none

    1)      CSPolicy: pol-cont-sw   CSVserver: vs-cont-sw   Priority: 100   Hits: 0

    1)      Policy : pol-ssl Priority:0
    2)      Policy : cf-pol Priority:1      Inherited
Done
<!--NeedCopy-->
```

Invoke a policy label or virtual server policy bank by using the GUI

1. Bind a policy, as described in Bind a policy globally, Bind a policy to a virtual server, or Bind a policy to a policy label. Alternatively, you can enter a NOPOLICY "dummy" entry instead of a policy name. You do this if you do not want to evaluate a policy before evaluating the policy bank.
2. In the Invoke field, select the name of the policy label or virtual server policy bank that you want to evaluate if traffic matches the bound policy. A message in the status bar indicates that the policy label or virtual server policy bank is invoked successfully.

Remove a policy label invocation by using the GUI

1. Open the policy and clear the Invoke field. Unbinding the policy also removes the invocation of the label. A message in the status bar indicates that the policy label is removed successfully.

Configure and bind policies with the policy manager

September 21, 2020

> **Warning:**
>
> Classic policy expressions are no longer supported from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies.

Some applications provide a specialized Policy Manager in the NetScaler GUI to simplify configuring policy banks. It also lets you find and delete policies and actions that are not being used.

The Policy Manager is currently available for the Rewrite, Integrated Caching, Responder, and Compression features.

The following are keyboard equivalents for the procedures in this section:

- For editing a cell in the Policy Manager, you can tab to the cell and click F2 or press the SPACE bar on the keyboard.
- To select an entry in a drop-down menu, you can tab to the entry, press the space bar to view the drop-down menu, use the UP and DOWN ARROW keys to navigate to the entry that you want, and press the space bar again to select the entry.
- To cancel a selection in a drop-down menu, press the Escape key.
- To insert a policy, tab to the row above the insertion point and press Control + Insert, or click Insert Policy.
- To remove a policy, tab to the row that contains the policy and press Delete.

Note: Note that when you delete the policy, the NetScaler searches the Goto Expression values of other policies in the bank. If any of these Goto Expression values match the priority level of the deleted policy, they are removed.

Configure policy bindings by using the policy manager

1. In the navigation pane, click the feature for which you want to configure policies. The choices are Responder, Integrated Caching, Rewrite or Compression.

2. In the details pane, click Policy Manager.

3. If you are configuring classic policy bindings for compression, in the Compression Policy Manager dialog box, click **Switch to Classic Syntax**. The dialog box switches to the classic syntax view and displays the Switch to Advanced Policy button. At any time before you complete configuring policy bindings, if you want to configure bindings for policies that use the Advanced Policy click the Switch to Advanced Policy button.

4. For features other than Responder, to specify the bind point, click Request or Response, and then click one of the request-time or response-time bind points. The options are Override Global, LB Virtual Server, CS Virtual Server, Default Global, or Policy Label. If you are configuring the Responder, the Request and Response flow types are not available.

5. To bind a policy to this bind point, click Insert Policy, and select a previously configured policy, a NOPOLICY label, or the New policy option. Depending on the option that you select, you have the following choices:

   - **New policy:** Create the policy as described in "Create or modify a policy," and then configure the priority level, GoTo expression, and policy invocation as described in the table, "Format of each entry in a policy bank."
   - **Existing policy**, **NOPOLICY**, or **NOPOLICY<feature name>**: Configure the priority level, GoTo expression, and policy invocation as described in the table, "Format of each entry in

a policy bank.” The **NOPOLICY** or **NOPOLICY<feature name>** options are available only for policies that use Advanced Policies.

6. Repeat the preceding steps to add entries to this policy bank.
7. To modify the priority level for an entry, you can do any of the following:
   - Double-click the Priority field for an entry and edit the value.
   - Click and drag a policy to another row in the table.
   - Click Regenerate Priorities.

   In all three cases, priority levels of all other policies are modified as needed to accommodate the new value. Goto Expressions with integer values are also updated automatically. For example, if you change a priority value of 10 to 100, all policies with a Goto Expression value of 10 are updated to the value 100.
8. To change the policy, action, or policy bank invocation for an row in the table, click the down arrow to the right of the entry and do one of the following:
   - To change the policy, select another policy name or select New Policy and follow the steps in Create or modify a policy.
   - To change the Goto Expression, select Next, End, USE_INVOCATION_RESULT, or select more and enter an expression whose result returns the priority level of another entry in this policy bank.
   - To modify an invocation, select an existing policy bank, or click New Policy Label and follow the steps in Bind a policy to a policy label.
9. To unbind a policy or a policy label invocation from this bank, click any field in the row that contains the policy or policy label, and then click Unbind Policy.
10. When you are done, click Apply Changes. A message in the status bar indicates that the policy is bound successfully.

Remove unused policies by using the policy manager

1. In the navigation pane, click the feature for which you want to configure the policy bank. The choices are Responder, Integrated Caching, or Rewrite.
2. In the details pane, click <Feature Name> policy manager.
3. In the <**Feature Name**> **Policy Manage**r dialog box, click **Cleanup Configuration**.
4. In the **Cleanup Configuration** dialog box, select the items that you want to delete, and then click **Remove**.
5. In the Remove dialog box, click **Yes**.
6. Click **Close**. A message in the status bar indicates that the policy is removed successfully.

Configuring advanced policy expression: getting started

September 21, 2020

Advanced policies evaluate data on the basis of information that you supply in Advanced policy expressions. An Advanced policy expression analyzes data elements (for example, HTTP headers, source IP addresses, the NetScaler system time, and POST body data). In addition to configuring an Advanced policy expression in a policy, in some NetScaler features, you configure Advanced policy expression outside of the context of a policy.

To create an Advanced policy expression, you select a prefix that identifies a piece of data that you want to analyze, and then you specify an operation to perform on the data. For example, an operation can match a piece of data with a text string that you specify, or it can transform a text string into an HTTP header. Other operations match a returned string with a set of strings or a string pattern. You configure compound expressions by specifying Boolean and arithmetic operators, and by using parentheses to control the order of evaluation.

Advanced policy expression can also contain classic expressions. You can assign a name to a frequently used expression to avoid having to build the expression repeatedly.

Policies and a few other entities include rules that the NetScaler uses to evaluate a packet in the traffic flowing through it, to extract data from the NetScaler system itself, to send a request (a “callout”) to an external application, or to analyze another piece of data. A rule takes the form of a logical expression that is compared against traffic and ultimately returns values of TRUE or FALSE.

The elements of the rule can themselves return TRUE or FALSE, string, or numeric values.

Before configuring an Advanced policy expression, you need to understand the characteristics of the data that the policy or other entity is to evaluate. For example, when working with the Integrated Caching feature, a policy determines what data can be stored in the cache. With Integrated Caching, you need to know the URLs, headers, and other data in the HTTP requests and responses that the NetScaler receives. With this knowledge, you can configure policies that match the actual data and enable the NetScaler to manage caching for HTTP traffic. This information helps you determine the type of expression that you need to configure in the policy.

Basic elements of an advanced policy expression

September 21, 2020

An Advanced policy expression consists of, at a minimum, a prefix (or a single element used in place of a prefix). Most expressions also specify an operation to be performed on the data that the prefix identifies. You format an expression of up to 1,499 characters as follows:

`<prefix>.<operation> [<compound-operator> <prefix>.<operation>. . .]`

where

- <prefix>

  is an anchor point for starting an expression.

  The prefix is a period-delimited key that identifies a unit of data. For example, the following prefix examines HTTP requests for the presence of a header named Content-Type:

  http.req.header("Content-Type")

  Prefixes can also be used on their own to return the value of the object that the prefix identifies.

- <operation>

  identifies an evaluation that is to be performed on the data identified by the prefix.

  For example, consider the following expression:

  http.req.header("Content-Type").eq("text/html")

  In this expression, the following is the operator component:

  eq("text/html")

  This operator causes the NetScaler to evaluate any HTTP requests that contain a Content-Type header, and in particular, to determine if the value of this header is equal to the string "text/html." For more information, see Operations.

- <compound-operator>

  is a Boolean or arithmetic operator that forms a compound expression from multiple prefix or prefix.operation elements.

  For example, consider the following expression:

  http.req.header("Content-Type").eq("text/html") && http.req.url.contains(".html")

Prefixes

An expression prefix represents a discrete piece of data. For example, an expression prefix can represent an HTTP URL, an HTTP Cookie header, or a string in the body of an HTTP POST request. An expression prefix can identify and return a wide variety of data types, including the following:

- A client IP address in a TCP/IP packet
- NetScaler system time
- An external callout over HTTP
- A TCP or UDP record type

In most cases, an expression prefix begins with one of the following keywords:

- CLIENT:
    - Identifies a characteristic of the client that is either sending a request or receiving a response, as in the following examples:
    - The prefix client.ip.dst designates the destination IP address in the request or response.
    - The prefix client.ip.src designates the source IP address.
- HTTP:
    - Identifies an element in an HTTP request or a response, as in the following examples:
    - The prefix http.req.body(integer) designates the body of the HTTP request as a multiline text object, up to the character position designated in integer.
    - The prefix http.req.header("header_name") designates an HTTP header, as specified in header_name.
    - The prefix http.req.url designates an HTTP URL in URL-encoded format.
- SERVER:

  Identifies an element in the server that is either processing a request or sending a response.
- SYS:

  Identifies a characteristic of the NetScaler that is processing the traffic.

  Note: Note that DNS policies support only SYS, CLIENT, and SERVER objects.

  In addition, in the NetScaler Gateway, the Clientless VPN function can use the following types of prefixes:
- TEXT:

  Identifies any text element in a request or a response.
- TARGET:

  Identifies the target of a connection.
- URL:

  Identifies an element in the URL portion of an HTTP request or response.

As a general rule of thumb, any expression prefix can be a self-contained expression. For example, the following prefix is a complete expression that returns the contents of the HTTP header specified in the string argument (enclosed in quotation marks):

```
http.res.header.("myheader")
```

Or you can combine prefixes with simple operations to determine TRUE and FALSE values. For example, the following returns a value of TRUE or FALSE:

```
http.res.header.("myheader").exists
```

You can also use complex operations on individual prefixes and multiple prefixes within an expression, as in the following example:

```
http.req.url.length + http.req.cookie.length <= 500
```

Which expression prefixes you can specify depends on the NetScaler feature. The following table describes the expression prefixes that are of interest on a per-feature basis

Feature	Types of Expression Prefix Used in the Feature
DNS	SYS, CLIENT, SERVER
Responder in Protection Features	HTTP, SYS, CLIENT
Content Switching	HTTP, SYS, CLIENT
Rewrite	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN
Integrated Caching	HTTP, SYS, CLIENT, SERVER
NetScaler Gateway, Clientless Access	HTTP, SYS, CLIENT, SERVER, URL, TEXT, TARGET, VPN

Table 1. Permitted Types of Expression Prefixes in Various NetScaler Features

Note: For details on the permitted expression prefixes in a feature, see the documentation for that feature.

Single-element expressions

The simplest type of Advanced policy expression contains a single element. This element can be one of the following:

- true. An Advanced policy expression can consist simply of the value true. This type of expression always returns a value of TRUE. It is useful for chaining policy actions and triggering Goto expressions.
- false. An Advanced policy expression can consist simply of the value false. This type of expression always returns a value of FALSE.
- A prefix for a compound expression. For example, the prefix HTTP.REQ.HOSTNAME is a complete expression that returns a host name and HTTP.REQ.URL is a complete expression that returns a URL. The prefix could also be used in conjunction with operations and additional prefixes to form a compound expression.

Operations

In most expressions, you also specify an operation on the data that the prefix identifies. For example, suppose that you specify the following prefix:

```
http.req.url
```

This prefix extracts URLs in HTTP requests. This expression prefix does not require any operators to be used in an expression. However, when you configure an expression that processes HTTP request URLs, you can specify operations that analyze particular characteristics of the URL. Following are a few possibilities:

- Search for a particular host name in the URL.
- Search for a particular path in the URL.
- Evaluate the length of the URL.
- Search for a string in the URL that indicates a time stamp and convert it to GMT.

The following is an example of a prefix that identifies an HTTP header named Server and an operation that searches for the string IIS in the header value:

```
http.res.header("Server").contains("IIS")
```

Following is an example of a prefix that identifies host names and an operation that searches for the string "www.mycompany.com" as the value of the name:

```
http.req.hostname.eq("www.mycompany.com")
```

Basic operations on expression prefixes

The following table describes a few of the basic operations that can be performed on expression prefixes.

Operation	Determines Whether or Not
CONTAINS(<string>)	The object matches <string>. Following is an example: http.req.header("Cache-Control").contains("no-cache")
EXISTS	A particular item is present in an object. Following is an example: http.res.header("MyHdr").exists
EQ(<text>)	A particular non-numeric value is present in an object. Following is an example: http.req.method.eq(post)

Operation	Determines Whether or Not
EQ(<integer>)	A particular numeric value is present in an object. Following is an example: client.ip.dst.eq(10.100.10.100)
LT(<integer>)	An object's value is less than a particular value. Following is an example: http.req.content_length.lt(5000)
GT(<integer>)	An object's value is greater than a particular value. Following is an example: http.req.content_length.gt(5)

The following table summarizes a few of the available types of operations.

Operation Type	Description
Text operations	Match individual strings and sets of strings with any portion of a target. The target can be an entire string, the start of a string, or any portion of text in between the start and the end of the string. For example, you can extract the string "XYZ" from "XYZSomeText". Or, you can compare an HTTP header value with an array of different strings. You can also transform text into another type of data. Following are examples: Transform a string into an integer value, create a list from the query strings in a URL, and transform a string into a time value.
Numeric operations	Numeric operations include applying arithmetic operators, evaluating content length, the number of items in a list, dates, times, and IP addresses.

Compound advanced policy expressions

July 22, 2021

You can configure an Advanced policy expression that contains Boolean or arithmetic operators and multiple atomic operations. The following compound expression contains a boolean AND:

```
http.req.hostname.eq("mycompany.com")&& http.req.method.eq(post)
```

The following expression adds the value of two targets, and compares the result to a third value:

```
http.req.url.length + http.req.cookie.length \<= 500
```

A compound expression can contain any number of logical and arithmetic operators. The following expression evaluates the length of an HTTP request on the basis of its URL and cookie, evaluates text in the header, and performs a Boolean AND on these two results:

```
http.req.url.length + http.req.cookie.length \<= 500 && http.req.header.
contains("some text")
```

You can use parentheses to control the order of evaluation in a compound expression.

Booleans in compound expressions

You configure compound expressions with the following operators:

- &&.

  This operator is a logical AND. For the expression to evaluate to TRUE, all components that are joined by the And must evaluate to TRUE. Following is an example:

  http.req.url.hostname.eq("myHost") && http.req.header("myHeader").exists

.

- This operator is a logical OR. If any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE.
- !.

  Performs a logical NOT on the expression.

In some cases, the NetScaler GUI offers AND, NOT, and OR operators in the Add Expression dialog box. However, these are of limited use. Citrix recommends that you use the operators &&, , and ! to configure compound expressions that use Boolean logic.

Parentheses in compound expressions

You can use parentheses to control the order of evaluation of an expression. The following is an example:

```
http.req.url.contains("myCompany.com")|| (http.req.url.hostname.eq("myHost"
)&& http.req.header("myHeader").exists)
```

The following is another example:

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type").
eq("text/html"))|| (http.req.header("Transfer-Encoding").exists || http.req
.header("Content-Length").exists)
```

Compound operations for strings

The following table describes operators that you can use to configure compound operations on string data.

Operations that produce a string value	Description
str + str	Concatenates the value of the expression on the left of the operator with the value on the right. Following is an example:http.req.hostname + http.req.url.protocol
str + num	Concatenates the value of the expression on the left of the operator with a numeric value on the right. Following is an example:http.req.hostname + http.req.url.content_length
num + str	Concatenates the numeric value of the expression on the left side of the operator with a string value on the right. Following is an example:http.req.url.content_length + http.req.url.hostname
str + ip	Concatenates the string value of the expression on the left side of the operator with an IP address value on the right. Following is an example:http.req.hostname + 10.00.000.00

Operations that produce a string value	Description
ip + str	Concatenates the IP address value of the expression on the left of the operator with a string value on the right.Following is an example:client.ip.dst + http.req.url.hostname
str1 ALT str2	.Uses string2 if the evaluation of string1 results in an undef exception or the result is a null string; otherwise uses string1 and never evaluates string2 Following is an example:http.req.hostname alt client.ip.src

Operations on strings that produce a result of TRUE or FALSE	Description
str == str	Evaluates whether the strings on either side of the operator are the same. Following is an example: http.req.header("myheader") == http.res.header("myheader")
str <= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or precedes it alphabetically.
str >= str	Evaluates whether the string on the left side of the operator is the same as the string on the right, or follows it alphabetically.
str < str	Evaluates whether the string on the left side of the operator precedes the string on the right alphabetically.
str > str	Evaluates whether the string on the left side of the operator follows the string on the right alphabetically.
str !!= str	Evaluates whether the strings on either side of the operator are different.

Logical operations on strings	Description	
bool && bool	This operator is a logical AND. When evaluating the components of the compound expression, all components that are joined by the AND must evaluate to TRUE. Following is an example:http.req.method.eq && http.req.url.query.conta && my_pagelabel”)	
bool || bool	This operator is a logical OR. When evaluating the components of the compound expression, if any component of the expression that is joined by the OR evaluates to TRUE, the entire expression is TRUE. Following is an example:http.req.url.contains(“.js”)	http.res.header.(“Content-Type”).contains(“javascript”)
bool	Performs a logical NOT on the expression.	

Compound operations for numbers

You can configure compound numeric expressions. For example, the following expression returns a numeric value that is the sum of an HTTP header length and a URL length:

`http.req.header.length + http.req.url.length`

The following tables describes operators that you can use to configure compound expressions for numeric data.

Arithmetic operations on numbers	Description
num + num	Add the value of the expression on the left of the operator to the value of the expression on the right. Following is an example:http.req.content_le + http.req.url.length
num – num	Subtract the value of the expression on the right of the operator from the value of the expression on the left.
num x num	Multiply the value of the expression on the left of the operator with the value of the expression on the right. Following is an example:client.interface.rxth * 9
num / num	Divide the value of the expression on the left of the operator by the value of the expression on the right.

Arithmetic operations on numbers	Description
num % num	Calculate the modulo, or the numeric remainder on a division of the value of the expression on the left of the operator by the value of the expression on the right. For example, the values “15 mod 4” equals 3, and “12 mod 4” equals 0.
~number	Returns a number after applying a bitwise logical negation of the number. The following example assumes that numeric.expression returns 12 (binary 1100):~numeric.expression. The result of applying the ~ operator is -11 (a binary 1110011, 32 bits total with all ones to the left). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
number ^ number	Compares two bit patterns of equal length and performs an XOR operation on each pair of corresponding bits in each number argument, returning 1 if the bits are different, and 0 if they are the same. Returns a number after applying a bitwise XOR to the integer argument and the current number value. If the values in the bitwise comparison are the same, the returned value is a 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010):numeric.expression1 ^ numeric.expression2The result of applying the ^ operator to the entire expression is 6 (binary 0110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description		
number | number	Returns a number after applying a bitwise OR to the number values. If either value in the bitwise comparison is a 1, the returned value is a 1. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010):numeric.expression1	numeric.expression2The result of applying the	operator to the entire expression is 14 (binary 1110). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
number & number	Compares two bit patterns of equal length and performs a bitwise AND operation on each pair of corresponding bits, returning 1 if both of the bits contains a value of 1, and 0 if either bits are 0. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 10 (binary 1010):numeric.expression1 & numeric.expression2. The whole expression evaluates to 8 (binary 1000). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
num « num	Returns a number after a bitwise left shift of the number value by the right-side number argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3: numeric.expression1 « numeric.expression2 The result of applying the LSHIFT operator is 96 (a binary 1100000).Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Arithmetic operations on numbers	Description
num » num	Returns a number after a bitwise right shift of the number value by the integer argument number of bits. Note that the number of bits shifted is integer modulo 32. The following example assumes that numeric.expression1 returns 12 (binary 1100) and numeric.expression2 returns 3:numeric.expression1 » numeric.expression2 The result of applying the RSHIFT operator is 1 (a binary 0001). Note that all returned values of less than 32 bits before applying the operator implicitly have zeros to the left to make them 32 bits wide.

Numeric operators that produce a result of TRUE or FALSE	Description
num == num	Determine if the value of the expression on the left of the operator is equal to the value of the expression on the right.

Numeric operators that produce a result of TRUE or FALSE	Description
num != num	Determine if the value of the expression on the left of the operator is not equal to the value of the expression on the right.
num > num	Determine if the value of the expression on the left of the operator is greater than the value of the expression on the right.
num < num	Determine if the value of the expression on the left of the operator is less than the value of the expression on the right.
num >= num	Determine if the value of the expression on the left of the operator is greater than or equal to the value of the expression on the right.
num <= num	Determine if the value of the expression on the left of the operator is less than or equal to the value of the expression on the right

Functions for data types in the policy infrastructure

The NetScaler policy infrastructure supports the following numeric data types:

- Integer (32 bits)
- Unsigned long (64 bits)
- Double (64 bits)

Simple expressions can return all of these data types. Therefore, you can create compound expressions that use arithmetic operators and logical operators to evaluate or return values of these data types. Additionally, you can use all of these values in policy expressions. Literal constants of type unsigned long can be specified by appending the string ul to the number. Literal constants of type double contain a period (.), an exponent, or both.

Arithmetic Operators, Logical Operators, and Type Promotion

In compound expressions, the following standard arithmetic and logical operators can be used for the double and unsigned long data types:

- +, -, *, and /

%, ~, ^, &,	, <<, and >> (do not apply to double)

-
- ==, !=, >, <, >=, and <=

All of these operators have the same meaning as in the C programming language.

In all cases of mixed operations between operands of type integer, unsigned long, and double, type promotion is performed so that the operation can be performed on operands of the same type. A type of lower precedence is automatically promoted to the type of the operand with the highest precedence involved in the operation. The order of precedence (higher to lower) is as follows:

- Double
- Unsigned long
- Integer

Therefore, an operation that returns a numeric result returns a result of the highest type involved in the operation.

For example, if the operands are of type integer and unsigned long, the integer operand is automatically converted to type unsigned long. This type conversion is performed even in simple expressions in which the type of data identified by the expression prefix does not match the type of data that is passed as the argument to the function. To illustrate such an example, in the operation HTTP.REQ.CONTENT_LENGTH.DIV(3ul), the integer returned by the prefix HTTP.REQ.CONTENT_LENGTH is automatically converted to unsigned long (the type of the data passed as the argument to the DIV() function), and an unsigned long division is performed. Similarly, the argument can be promoted in an expression. For example, HTTP.REQ.HEADER("myHeader").TYPECAST_DOUBLE_AT.DIV(5) promotes the integer 5 to type double and performs double-precision division.

For information about expressions for casting data of one type to data of another type, see Typecasting data.

Specify the character set in expressions

September 21, 2020

The policy infrastructure on the Citrix® NetScaler® appliance supports the ASCII and UTF-8 character sets. The default character set is ASCII. If the traffic for which you are configuring an expression consists of only ASCII characters, you need not specify the character set in the expression. However, you must specify the character set in every simple expression that is meant for UTF-8 traffic. To specify the

UTF-8 character set in a simple expression, you must include the SET_CHAR_SET(<charset>) function, with <charset> specified as UTF_8, as shown in the following examples:

```
HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8).CONTAINS("ß")

HTTP.RES.BODY(100).SET_CHAR_SET(UTF_8).BEFORE_STR("Bücher").AFTER_STR("
    Wörterbuch")
<!--NeedCopy-->
```

In an expression, the SET_CHAR_SET() function must be introduced at the point in the expression after which data processing must be carried out in the specified character set. For example, in the expression HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).CONTAINS_ANY("Greek_ alphabet"), if the strings stored in the pattern set "Greek_alphabet" are in UTF-8, you must include the SET_CHAR_SET(UTF_8) function immediately before the CONTAINS_ANY("<string>") function, as follows:

`HTTP.REQ.BODY(1000).AFTER_REGEX(re/following example/).BEFORE_REGEX(re/In the preceding example/).SET_CHAR_SET(UTF_8).CONTAINS_ANY("Greek_ alphabet")`

The SET_CHAR_SET() function sets the character set for all further processing (that is, for all subsequent functions) in the expression unless it is overridden later in the expression by another SET_CHAR_SET() function that changes the character set. Therefore, if all the functions in a given simple expression are intended for UTF-8, you can include the SET_CHAR_SET(UTF_8) function immediately after functions that identify text (for example, the HEADER("<name>") or BODY(<int>) functions). In the second example that follows the first paragraph above, if the ASCII arguments passed to the AFTER_REGEX() and BEFORE_REGEX() functions are changed to UTF-8 strings, you can include the SET_CHAR_SET(UTF_8) function immediately after the BODY(1000) function, as follows:

`HTTP.REQ.BODY(1000).SET_CHAR_SET(UTF_8).AFTER_REGEX(re/Bücher/).BEFORE_REGEX(re/Wörterbuch/).CONTAINS_ANY("Greek_alphabet")`

The UTF-8 character set is a superset of the ASCII character set, so expressions configured for the ASCII character set continue to work as expected if you change the character set to UTF-8.

Compound expressions with different character sets

In a compound expression, if one subset of expressions is configured to work with data in the ASCII character set and the rest of the expressions are configured to work with data in the UTF-8 character set, the character set specified for each individual expression is considered when the expressions are evaluated individually. However, when processing the compound expression, just before processing the operators, the appliance promotes the character set of the returned ASCII values to UTF-8. For example, in the following compound expression, the first simple expression evaluates data in the ASCII character set while the second simple expression evaluates data in the UTF-8 character set:

```
HTTP.REQ.HEADER("MyHeader")== HTTP.REQ.BODY(10).SET_CHAR_SET(UTF_8)
```

However, when processing the compound expression, just before evaluating the “is equal to” Boolean operator, the NetScaler appliance promotes the character set of the value returned by HTTP.REQ.HEADER(“MyHeader”) to UTF-8.

The first simple expression in the following example evaluates data in the ASCII character set. However, when the NetScaler appliance processes the compound expression, just before concatenating the results of the two simple expressions, the appliance promotes the character set of the value returned by HTTP.REQ.BODY(10) to UTF-8.

```
HTTP.REQ.BODY(10)+ HTTP.REQ.HEADER("MyHeader").SET_CHAR_SET(UTF_8)
```

Consequently, the compound expression returns data in the UTF-8 character set.

Specify the character set based on the character set of traffic

You can set the character set to UTF-8 on the basis of traffic characteristics. If you are not sure whether the character set of the traffic being evaluated is UTF-8, you can configure a compound expression in which the first expression checks for UTF-8 traffic and subsequent expressions set the character set to UTF-8. Following is an example of a compound expression that first checks the value of “charset” in the request’s Content-Type header for “UTF-8” before checking whether the first 1000 bytes in the request contain the UTF-8 string Bücher:

```
HTTP.REQ.HEADER("Content-Type").SET_TEXT_MODE(IGNORECASE).TYPECAST_NVLIST_T
('=', '; ', '"').VALUE("charset").EQ("UTF-8")&& HTTP.REQ.BODY(1000).SET_CHAR_SET
(UTF_8).CONTAINS("Bücher")
```

If you are sure that the character set of the traffic being evaluated is UTF-8, the second expression in the example is sufficient.

Character and string literals in expressions

During expression evaluation, even if the current character set is ASCII, character literals and string literals, which are enclosed in single quotation marks (‘’) and quotation marks (“”), respectively, are considered to be literals in the UTF-8 character set. In a given expression, if a function is operating on character or string literals in the ASCII character set and you include a non-ASCII character in the literal, an error is returned.

Values in hexadecimal and octal formats

When configuring an expression, you can enter values in octal and hexadecimal formats. However, each hexadecimal or octal byte is considered a UTF-8 byte. Invalid UTF-8 bytes result in errors regardless of whether the value is entered manually or pasted from the clipboard. For example, “\xce\x20” is an invalid UTF-8 character because “c8” cannot be followed by “20” (each byte in a multi-byte UTF-8 string must have the high bit set). Another example of an invalid UTF-8 character is “\xce \xa9,” since the hexadecimal characters are separated by a white-space character.

Functions that return UTF-8 strings

Only the
<text>.XPATH and
<text>.XPATH_JSON functions always return UTF-8 strings. The following MYSQL routines determine at runtime which character set to return, depending on the data in the protocol:

- MYSQL_CLIENT_T.USER
- MYSQL_CLIENT_T.DATABASE
- MYSQL_REQ_QUERY_T.COMMAND
- MYSQL_REQ_QUERY_T.TEXT
- MYSQL_REQ_QUERY_T.TEXT(<unsigned int>)
- MYSQL_RES_ERROR_T.SQLSTATE
- MYSQL_RES_ERROR_T.MESSAGE
- MYSQL_RES_FIELD_T.CATALOG
- MYSQL_RES_FIELD_T.DB
- MYSQL_RES_FIELD_T.TABLE
- MYSQL_RES_FIELD_T.ORIGINAL_TABLE
- MYSQL_RES_FIELD_T.NAME
- MYSQL_RES_FIELD_T.ORIGINAL_NAME
- MYSQL_RES_OK_T.MESSAGE
- MYSQL_RES_ROW_T.TEXT_ELEM(<unsigned int>)

Terminal connection settings for UTF-8

When you set up a connection to the NetScaler appliance by using a terminal connection (by using PuTTY, for example), you must set the character set for transmission of data to UTF-8.

Classic expressions in advanced policy expressions

September 21, 2020

> **Warning**
>
> Classic policy expressions are no longer supported from NetScaler 12.0 build 56.20 onwards and

> as an alternative, Citrix recommends you to use Advanced policies. For more information, see Configure advanced policy expressions: Get started.

Classic expressions describe basic characteristics of traffic. In some cases, you may want to use a classic expression in an Advanced policy expression are no longer supported from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies topic. You can do so with the Advanced policy expression configuration tool. This can be helpful when manually migrating the older classic expressions to the Advanced policy.

Note that when you upgrade the NetScaler to version 9.0 or higher, Integrated Caching policies are automatically upgraded to Advanced policies, and the expressions in these policies are upgraded to the Advanced policies.

The following is the syntax for all Advanced policy expressions that use a classic expression:

SYS.EVAL_CLASSIC_EXPR("expression")

> **Note**
>
> The syntax and the metadata for the SYS.EVAL_CLASSIC_EXPR expression is getting deprecated. You can manually convert or use nspepi tool to convert the Classic expression to advanced expression.

Following are examples of the SYS.EVAL_CLASSIC_EXPR("expression") expression:

```
sys.eval_classic_expr("req.ssl.client.cipher.bits > 1000")
sys.eval_classic_expr("url contains abc")
sys.eval_classic_expr("req.ip.sourceip == 10.102.1.61 -netmask
    255.255.255.255")
sys.eval_classic_expr("time >= *:30:00GMT")
sys.eval_classic_expr("e1 || e2")
sys.eval_classic_expr("req.http.urllen > 50")
sys.eval_classic_expr("dayofweek == wedGMT")
<!--NeedCopy-->
```

Configure advanced policy expressions in a policy

September 21, 2020

You can configure an Advanced policy expression of up to 1,499 characters in a policy. The user interface for Advanced policy expressions depends to some extent on the feature for which you are configuring the expression, and on whether you are configuring an expression for a policy or for another use.

When configuring expressions on the command line, you delimit the expression by using quotation marks (“. . .” or ‘. . .’). Within an expression, you escape additional quotation marks by using a back-slash (). For example, the following are standard methods for escaping quotation marks in an expression:

`"\"abc\""`

‘`\"abc\"`’

You must also use a backslash to escape question marks and other backslashes on the command line. For example, the expression http.req.url.contains(“\?”) requires a backslash so that the question mark is parsed. Note that the backslash character will not appear on the command line after you type the question mark. On the other hand, if you escape a backslash (for example, in the expression ‘http.req.url.contains(“\\http”)’), the escape characters are echoed on the command line.

To make an entry more readable, you can escape the quotation marks for an entire expression. At the start of the expression you enter the escape sequence “q” plus one of the following special characters: /{<	~$^+=&%@‘?.

You enter only the special character at the end of the expression, as follows:

```
q@http.req.url.contains("sometext") && http.req.cookie.exists@

q~http.req.url.contains("sometext") && http.req.cookie.exists~
<!--NeedCopy-->
```

Note that an expression that uses the { delimiter is closed with }.

For some features (for example, Integrated Caching and Responder), the policy configuration dialog box provides a secondary dialog box for configuring expressions. This dialog enables you to choose from drop-down lists that show the available choices at each point during expression configuration. You cannot use arithmetic operators when using these configuration dialogs, but most other advanced policy expression features are available. To use arithmetic operators , write your expressions in free-form format.

Configure an Advanced policy syntax rule by using the CLI

> **Note:**
>
> Default syntax policy is now renamed as Advanced policy.

At the command prompt, type the following commands to configure a default syntax rule and verify the configuration:

1. `add cache|dns|rewrite|cs policy policyName **-rule** expression featureSpecificPa`
   `**-action**`
2. `show cache|dns|rewrite|cs policy policyName`
   Following is an example of configuring a caching policy:

   Example:

   ```
   > add cache policy pol-cache -rule http.req.content_length.le(5) -action INVAL
    Done

   > show cache policy pol-cache
           Name: pol-cache
           Rule: http.req.content_length.le(5)
           CacheAction: INVAL
           Invalidate groups: DEFAULT
           UndefAction: Use Global
           Hits: 0
           Undef Hits: 0

    Done
   <!--NeedCopy-->
   ```

Configure a default syntax policy expression by using the GUI

1. In the navigation pane, click the name of the feature where you want to configure a policy, for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching, and then click **Policies**.
2. Click Add.
3. For most features, click in the **Expression** field. For content switching, click **Configure**.
4. Click the **Prefix** icon (the house) and select the first expression prefix from the drop-down list. For example, in Responder, the options are HTTP, SYS, and CLIENT. The next set of applicable options appear in a drop-down list.
5. Double-click the next option to select it, and then type a period (.). Again, a set of applicable options appears in another drop-down list.
6. Continue selecting options until an entry field (signalled by parentheses) appears. When you see an entry field, enter an appropriate value in the parentheses. For example, if you select

GT(int) (greater-than, integer format), you specify an integer in the parentheses. Text strings are delimited by quotation marks. Following is an example:

`HTTP.REQ.BODY(1000).BETWEEN("this","that")`

To insert an operator between two parts of a compound expression, click the Operators icon (the sigma), and select the operator type. Following is an example of a configured expression with a Boolean OR (signalled by double vertical bars,	):

7. `HTTP.REQ.URL.EQ("www.mycompany.com")||HTTP.REQ.BODY(1000).BETWEEN("this","that")`
8. To insert a named expression, click the down arrow next to the Add icon (the plus sign) and select a named expression.
9. To configure an expression using drop-down menus, and to insert built-in expressions, click the Add icon (the plus sign). The **Add Expression** dialog box works in a similar way to the main dialog box, but it provides drop-down lists for selecting options, and it provides text fields for data entry instead of parentheses. This dialog box also provides a Frequently Used Expressions drop-down list that inserts commonly used expressions. When you are done adding the expression, click **OK**.
10. When finished, click **Create**. A message in the status bar indicates that the policy expression is configured successfully.

Test a default syntax expression by using the GUI

1. In the navigation pane, click the name of the feature for which you want to configure a policy (for example, you can select Integrated Caching, Responder, DNS, Rewrite, or Content Switching), and then click Policies.
2. Select a policy and click **Open**.
3. To test the expression, click the Evaluate icon (the check mark).
4. In the expression evaluator dialog box, select the Flow Type that matches the expression.
5. In the **HTTP Request Data** or **HTTP Response Data** field, paste the HTTP request or response that you want to parse with the expression, and click **Evaluate**. Note that you must supply a complete HTTP request or response, and the header and body should be separated by blank

line. Some programs that trap HTTP headers do not also trap the response. If you are copying and pasting only the header, insert a blank line at the end of the header to form a complete HTTP request or response.

6. Click **Close** to close this dialog box.

Configure named advanced policy expressions

September 21, 2020

Instead of retyping the same expression multiple times in multiple policies, you can configure a named expression and refer to the name any time you want to use the expression in a policy. For example, you could create the following named expressions:

- ThisExpression:

  ```
  http.req.body(100).contains("this")
  ```

- ThatExpression:

  ```
  http.req.body(100).contains("that")
  ```

You can then use these named expressions in a policy expression. For example, the following is a legal expression based on the preceding examples:

ThisExpression	ThatExpression

You can use the name of an advanced policy expression as the prefix to a function. The named expression can be either a simple expression or a compound expression. The function must be one that can operate on the type of data that is returned by the named expression.

Example 1: Simple Named Expression as a Prefix

The following simple named expression, which identifies a text string, can be used as a prefix to the AFTER_STR("<string>")function, which works with text data:

```
HTTP.REQ.BODY(1000)
```

If the name of the expression is top1KB, you can use top1KB.AFTER_STR("username") instead of HTTP.REQ.BODY(1000).AFTER_STR("username").

Example 2: Compound named expression as a prefix

You can create a compound named expression called basic_header_value to concatenate the user name in a request, a colon (:), and the user's password, as follows:

```
add policy expression basic_header_value "HTTP.REQ.USER.NAME + \":\" + HTTP.REQ.USER.PASSWD"
```

You can then use the name of the expression in a rewrite action, as shown in the following example:

```
add rewrite action insert_b64encoded_authorization insert_http_header authorization '"Basic " + basic_header_value.b64encode'-bypassSafetyCheck YES
```

In the example, in the expression that is used to construct the value of the custom header, the B64 encoding algorithm is applied to the string returned by the compound named expression.

You can also use a named expression (either by itself or as a prefix to a function) to create the text expression for the replacement target in a rewrite.

Configure a named default syntax expression by using the CLI

At the command prompt, type the following commands to configure a named expression and verify the configuration:

```
-  add policy expression \<name\>\<value\>

-  show policy expression \<name\>
<!--NeedCopy-->
```

Example:

```
> add policy expression myExp "http.req.body(100).contains(\"the other\")"
Done

> show policy expression myExp
    1)      Name: myExp  Expr: "http.req.body(100).contains("the other")"  Hits: 0 Type : ADVANCED
Done
<!--NeedCopy-->
```

The expression can be up to 1,499 characters.

Configure a named expression by using the GUI

1. In the navigation pane, expand **AppExpert**, and then click **Expressions**.
2. Click **Advanced Expressions**.
3. Click **Add**.
4. Enter a name and a description for the expression.
5. Configure the expression by using the process described in Configure advanced policy expression. A message in the status bar indicates that the policy expression is configured successfully.

Configure advanced policy expressions outside the context of a policy

September 21, 2020

A number of functions, including the following, can require an advanced policy expression that is not part of a policy:

- Integrated Caching selectors:

  You define multiple non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND relationship with the others.

- Load Balancing:

  You configure an expression for the TOKEN method of load balancing for a load balancing virtual server.

- Rewrite actions:

  Expressions define the location of the rewrite action and the type of rewriting to be performed, depending on the type of rewrite action that you are configuring. For example, a DELETE action only uses a target expression. A REPLACE action uses a target expression and an expression to configure the replacement text.

- Rate-based policies:

  You use advanced policy xpressions to configure Limit Selectors. You can use these selectors when configuring policies to throttle the rate of traffic to various servers. You define up to five non-compound expressions (selectlets) in the definition of the selector. Each selectlet is in an implicit logical AND with the others.

Configure an advanced policy expression outside a policy by using the CLI (cache selector example)

At the command prompt, type the following commands to configure an advanced policy expression outside a policy and verify the configuration:

```
-  add cache selector \<selectorName\> \<rule\>
-  show cache selector \<selectorName\>
<!--NeedCopy-->
```

Example:

```
> add cache selector mainpageSelector "http.req.cookie.value(\"ABC_def
  \")"
   "http.req.url.query.value(\"_ghi\")"selector "mainpageSelector"
       added
Done
> show cache selector mainpageSelector
            Name: mainpageSelector
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

Following is an equivalent command that uses the more readable q delimiter, as described in Configure advanced policy expressions in a policy:

```
> add cache selector mainpageSelector2 q~http.req.cookie.value("ABC_def
  ")~
   q~http.req.url.query.value("_ghi")~selector "mainpageSelector2"
       added
Done
> show cache selector mainpageSelector2
            Name: mainpageSelector2
            Expressions:
                    1) http.req.cookie.value("ABC_def")
                    2) http.req.url.query.value("_ghi")
Done
<!--NeedCopy-->
```

Advanced policy expressions: evaluating text

September 21, 2020

You can configure a policy with an advanced policy expression that evaluates text in a request or response. Advanced policy text expressions can range from simple expressions that perform string matching in HTTP headers to complex expressions that encode and decode text. You can configure text expressions to be case sensitive or case insensitive and to use or ignore spaces. You can also configure complex text expressions by combining text expressions with Boolean operators

You can use expression prefixes and operators for evaluating HTTP requests, HTTP responses, and VPN and Clientless VPN data. However, text expression prefixes are not restricted to evaluating these elements of your traffic.

About text expressions

September 21, 2020

You can configure various expressions for working with text that flows through the NetScaler appliance. Following are some examples of how you can parse text by using a default syntax expression:

- Determine that a particular HTTP header exists.

  For example, you may want to identify HTTP requests that contains a particular Accept-Language header for the purpose of directing the request to a particular server.

- Determine that a particular HTTP URL contains a particular string.

  For example, you may want to block requests for particular URLs. Note that the string can occur at the beginning, middle, or end of another string.

- Identify a POST request that is directed to a particular application.

  For example, you may want to identify all POST requests that are directed to a database application for the purpose of refreshing cached application data.

About operations on text

A text-based expression consists of at least one prefix to identify an element of data and usually (although not always) an operation on that prefix. Text-based operations can apply to any part of a request or a response. Basic operations on text include various types of string matches.

For example, the following expression compares a header value with a string:

```
http.req.header("myHeader").contains("some-text")
```

Following expressions are examples of matching a file type in a request:

```
http.req.url.suffix.contains("jpeg")
```

```
http.req.url.suffix.eq("jpeg")
```

In the preceding examples, the contains operator permits a partial match and the eq operator looks for an exact match.

Other operations are available to format the string before evaluating it. For example, you can use text operations to strip out quotes and white spaces, to convert the string to all lowercase, or to concatenate strings.

> **Note**: Complex operations are available to perform matching based on patterns or to convert one type of text format to another type.

For more information, see the following topics:

- Pattern sets and data sets.
- Regular expressions.
- Typecasting data.

Compounding and precedence in text expressions

You can apply various operators to combine text prefixes or expressions. For example, the following expression concatenates the returned values of each prefix:

```
http.req.hostname + http.req.url
```

Following is an example of a compound text expression that uses a logical AND. Both components of this expression must be TRUE for a request to match the expression:

```
http.req.method.eq(post)&& http.req.body(1024).startswith("destination=")
```

Note: For more information on operators for compounding, see
Compound default syntax expressions.

Categories of text expressions

The primary categories of text expressions that you can configure are:

- Information in HTTP headers, HTTP URLs, and the POST body in HTTP requests.

  For more information, see Expression prefixes for text in HTTP requests and responses.

- Information regarding a VPN or a clientless VPN.

  For more information, see Expression prefixes for VPNs and clientless VPNs.

- TCP payload information.

  For more information about TCP payload expressions, see Default syntax expressions: Parsing HTTP, TCP, and UDP data.

- Text in a Secure Sockets Layer (SSL) certificate.

  For information about text expressions for SSL and SSL certificate data, see Default syntax expressions: Parsing SSL certificates and Expressions for SSL certificate dates.

> **Note:** Parsing a document body, such as the body of a POST request, can affect performance. You may want to test the performance impact of policies that evaluate a document body.

Guidelines for text expressions

From a performance standpoint, it typically is best to use protocol-aware functions in an expression. For example, the following expression makes use of a protocol-aware function:

```
HTTP.REQ.URL.QUERY
```

The previous expression performs better than the following equivalent expression, which is based on string parsing:

```
HTTP.REQ.URL.AFTER_STR("?")
```

In the first case, the expression looks specifically at the URL query. In the second case, the expression scans the data for the first occurrence of a question mark.

There is also a performance benefit from structured parsing of text, as in the following expression:

```
HTTP.REQ.HEADER("Example").TYPECAST_LIST_T(',').GET(1)
```

(For more information on typecasting, seeTypecasting data. The typecasting expression, which collects comma-delimited data and structures it into a list, typically would perform better than the following unstructured equivalent:

```
HTTP.REQ.HEADER("Example").AFTER_STR(",").BEFORE_STR(",")
```

Finally, unstructured text expressions typically have better performance than regular expressions. For example, the following is an unstructured text expression:

```
HTTP.REQ.HEADER("Example").AFTER_STR("more")
```

The previous expression would generally provide better performance than the following equivalent, which uses a regular expression:

```
HTTP.REQ.HEADER("Example").AFTER_REGEX(re/more/)
```

For more information on regular expressions, see Regular expressions.

Expression prefixes for text in HTTP requests and responses

September 21, 2020

An HTTP request or response typically contains text, such as in the form of headers, header values, URLs, and POST body text. You can configure expressions to operate on one or more of these text-based items in an HTTP request or response.

Refer to the Prefix Expression table to configure and extract text from different parts of an HTTP request or response.

Expression prefixes for VPNs and clientless VPNs

September 21, 2020

The Advanced policy engine provides prefixes that are specific to parsing VPN or Clientless VPN data. This data includes the following:

- Host names, domains, and URLs in VPN traffic.
- Protocols in the VPN traffic.
- Queries in the VPN traffic.

These text elements are often URLs and components of URLs. In addition to applying the text-based operations on these elements, you can parse these elements by using operations that are specific to parsing URLs. For more information, see Expressions for extracting segments of URLs

The following table describes the expression prefixes for this type of data.

VPN and Clientless VPN Expression	Description
VPN.BASEURL.CVPN_DECODE	Extracts the original URL from a clientless VPN URL.
VPN.BASEURL.CVPN_ENCODE	Converts a URL to clientless VPN format.
VPN.BASEURL.HOSTNAME	Extracts the HTTP host name from the host name in the URL. This prefix cannot be used in bidirectional policies.

VPN and Clientless VPN Expression	Description
VPN.BASEURL.HOSTNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, this prefix extracts `mycompany.com`. This prefix returns incorrect results if the host name is an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix are case insensitive.
`VPN.BASEURL.HOSTNAME.EQ (\<hostname \>)`	Returns a Boolean TRUE if the host name matches `\<hostname\>`. The comparison is case insensitive. For example, if the host name is `www.mycompany.com`, the following returns TRUE: vpn.baseurl.hostname.eq(`"www.mycompany.com"`). If the text mode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.BASEURL.HOSTNAME.SERVER	Evaluates the server portion of the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the server is `www.mycompany.com`. All text operations after this prefix are case insensitive.
VPN.BASEURL.PATH	Extracts a slash- (/) separated list from the path component of the URL. For example, this prefix extracts /a/b/c/mypage.html from the following URL: <`http://www.mycompany.com/a/b/c/mypage.html?a=1`>. The following expression selects just the "a": http.req.url.path.get(1). For more information on the GET operation, see "Expressions for Extracting Segments of URLs."

VPN and Clientless VPN Expression	Description
VPN.BASEURL.PATH.IGNORE_EMPTY_ELEMENTS	This prefix ignores the elements in a list. For example, the following comma-separated list has an empty element after “a=10”: a=10,,b=11, ,c=89. The element following b=11 contains a space, and by default, is not considered an empty element. Consider the following HTTP header: Cust_Header : 123,,24, ,15. The following expression returns a count of 4 when evaluating this header: http.req.header(“Cust_Header”).typecase_list_t(‘,’).ig following expression returns a count of 5 when evaluating this header: http.req.header(“Cust_Header”).typecase_list_t(‘,’). count
VPN.BASEURL.PATH_AND_QUERY	Evaluates the text in the URL that follows the host name. For example, if the URL is `<http://www.mycompany.com/a/b/c/mypage.html?a=1>`, this prefix evaluates /a/b/c/mypage.html?a=1.
VPN.BASEURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
VPN.BASEURL.QUERY	Extracts a name-value list, using the “=” and “&” delimiters from the query string in a URL.
VPN.BASEURL.QUERY.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a name-value list. For example, in the following name-value list, there is an empty element following “a=10”: a=10;;b=11; ;c=89. The element following b=11 contains a space and is not considered an empty element. Consider the following HTTP header: Cust_Header : a=1;;b=2; ;c=3. The following expression produces a count of 4 after evaluating this header: http.req.header(“Cust_Header”).typecast_nvlist_t(‘=’, following expression produces a count of 5 after evaluating the header: http.req.header(“Cust_Header”).typecast_nvlist_t(‘=’,

VPN and Clientless VPN Expression	Description
VPN.BASEURL.SUFFIX	Evaluates the file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this operation selects "html."
VPN.CLIENTLESS_BASEURL	Evaluates the clientless VPN base URL.
VPN.CLIENTLESS_BASEURL.CVPN_DECODE	Extracts the original URL from the clientless VPN formatted URL.
VPN.CLIENTLESS_BASEURL.CVPN_ENCODE	Converts a URL to the clientless VPN format.
VPN.CLIENTLESS_BASEURL.HOSTNAME	Evaluates the host name in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_BASEURL.HOSTNAME.DOMAIN	Evaluates the domain name part of the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the domain is `mycompany.com`. This operation returns incorrect results if the host name is an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs.." All text operations after this prefix are case insensitive.
`VPN.CLIENTLESS_BASEURL.HOSTNAME.EQ(<hostname>)`	Returns a Boolean TRUE if the host name matches <hostname>. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the following is true: vpn.clientless_baseurl. hostname.eq(`"www.mycompany.com"`) The comparison is case insensitive. If the textmode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."

VPN and Clientless VPN Expression	Description
`VPN.CLIENTLESS_BASEURL.HOSTNAME.SERVER`	Evaluates the server part of a host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the server is `www.mycompany.com`. All text operations after this prefix are case insensitive.
VPN.CLIENTLESS_BASEURL.PATH	Evaluates a slash- (/) separated list in the URL path. For example, this prefix selects /a/b/c/mypage.html from the following URL: <`http://www.mycompany.com/a/b/c/mypage.html?a=1`>. The following expression selects "a" from the preceding URL: `http.req.url.path.get(1)`. For more information on the GET operation, see "Expressions for Extracting Segments of URLs."
`VPN.CLIENTLESS_BASEURL.PATH.IGNORE_EMPTY_ELEMENTS`	Ignores empty elements in a list. For example, if the list delimiter is a comma (,) the following list has an empty element following "a=10": a=10,b=11, ,c=89. The element following b=11 contains a space and is not considered an empty element.. Consider the following HTTP header: Cust_Header : 123,,24, ,15. The following expression returns a value of 4 after evaluating this header: http.req.header("Cust_Header").typecast_list_t(',').ig following expression returns a value of 5 after evaluating this header: http.req.header("Cust_Header").typecast_list_t(',').
`VPN.CLIENTLESS_BASEURL.PATH_AND_QUERY`	Evaluates the text following the host name in a URL. For example, this prefix selects /a/b/c/mypage.html?a=1 from the following URL: <`http://www.mycompany.com/a/b/c/mypage.html?a=1`>
`VPN.CLIENTLESS_BASEURL.PROTOCOL`	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.

VPN and Clientless VPN Expression	Description
`VPN.CLIENTLESS_BASEURL.QUERY`	Extracts a name-value list that uses the delimiters “=” and “&” from a URL query string.
`VPN.CLIENTLESS_BASEURL.QUERY.IGNORE_EMPTY_ ELEMENTS`	Ignores empty elements in a name-value list. For example, the following list contains an empty element after “a=10”: a=10;;b=11; ;c=89. The element following b=11 contains a space and is not considered an empty element. As another example, consider the following http header: Cust_Header : a=1;;b=2; ;c=3. The following expression returns a value of 4 after evaluating the preceding header: http.req.header(“Cust_Header”).typecast_nvlist_t(‘=’ following expression returns a value of 5 after evaluating the preceding header: http.req.header(“Cust_Header”).typecast_nvlist_t(‘=’
`VPN.CLIENTLESS_BASEURL.SUFFIX`	Evaluates the file suffix in a URL. For example, if the URL path is /a/b/c/mypage.html then this operation selects html.
VPN.CLIENTLESS_HOSTURL	Selects the clientless VPN host URL.
VPN.CLIENTLESS_HOSTURL.CVPN_DECODE	Selects the original URL from the clientless VPN formatted URL.
VPN.CLIENTLESS_HOSTURL.CVPN_ENCODE	Converts a URL to clientless VPN format.
VPN.CLIENTLESS_HOSTURL.HOSTNAME	Extracts the host name in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_HOSTURL.HOSTNAME.DOMAIN	Extracts the domain name from the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the domain is `mycompany.com`. This operation returns incorrect results if the host name contains an IP address. For information on expressions for IP addresses, see “Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs.” All text operations after this prefix are case insensitive.

VPN and Clientless VPN Expression	Description
VPN.CLIENTLESS_HOSTURL.HOSTNAME.EQ(<ho	Results in Boolean TRUE if the host name matches the \<`hostname`\> argument. The comparison is case insensitive. For example, if the host name is `www.mycompany.com` or `www.mycompany.com`., the following expression returns TRUE: vpn.clilentless_hosturl. hostname.eq(`"www.mycompany.com"`). If the text mode is URLENCODED, the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.CLIENTLESS_HOSTURL.HOSTNAME.SERVER	Evaluates the server part of the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the server is `www.mycompany.com`.. The comparison is case insensitive, and all text operations after this method are case insensitive.
VPN.CLIENTLESS_HOSTURL.PATH	Evaluates a slash- (/) separated list on the path component of the URL. For example, consider the following URL: `<http://www.mycompany.com/a/b/c/mypage.html?a=1>`. This prefix selects /a/b/c/mypage.html from the preceding URL.

VPN and Clientless VPN Expression	Description
VPN.CLIENTLESS_HOSTURL.PATH.IGNORE_EMPTY_ELEMENTS	This method ignores the empty elements in a list. For example, if the delimiter in a list is “,” the following list contains an empty element after the entry “a=10”: a=10,b=11, ,c=89. The element following b=11 contains a space and is not considered an empty element. Consider the following header: Cust_Header : 123„24, ,15. The following expression returns a value of 4 for this header: http.req.header(“Cust_Header”).typecast_list_t(‘,’). ignore_empty_elements.count. The following expression returns a value of 5 for the same header: http.req.header(“Cust_Header”).typecast_list_t(‘,’).
`VPN.CLIENTLESS_HOSTURL.PATH_AND_QUERY`	Evaluates the portion of the URL that follows the host name. For example, consider the following URL: <`http://www.mycompany.com/a/b/c/mypage.html?a=1`>. This prefix returns /a/b/c/mypage.html?a=1 from the preceding URL.
VPN.CLIENTLESS_HOSTURL.PROTOCOL	Evaluates the protocol in the URL. Do not use this prefix in bidirectional policies.
VPN.CLIENTLESS_HOSTURL.QUERY	Extracts a name-value list, using the “=” and “&” delimiters from a URL query string.

VPN and Clientless VPN Expression	Description
VPN.CLIENTLESS_HOSTURL.QUERY.IGNORE_EMPTY_ELEMENTS	Ignores empty elements in a name-value list. For example, the following list uses a semicolon (;) delimiter. This list contains an empty element after "a=10": a=10;;b=11; ;c=89. In the preceding example, the element following b=11 is not considered an empty element. Consider the following header: Cust_Header : a=1;;b=2; ;c=3. The following expression returns a value of 4 after evaluating this header: http.req.header("Cust_Header").typecast_nvlist_t('=', following expression returns a value of 5 after evaluating the same header: http.req.header("Cust_Header").typecast_nvlist_t('=',
VPN.CLIENTLESS_HOSTURL.SUFFIX	Extracts a file name suffix in a URL. For example, if the path is /a/b/c/my.page.html, this prefix selects html.
VPN.HOST.DOMAIN	Extracts the domain name part of the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the domain is `mycompany.com`. This prefix returns incorrect results if the host name contains an IP address. For information on expressions for IP addresses, see "Default Syntax Expressions: IP and MAC Addresses, Throughput, VLAN IDs." All text operations after this prefix case insensitive.

VPN and Clientless VPN Expression	Description
VPN.HOST.EQ(<hostname>)	Returns a Boolean TRUE value if the host name matches the <hostname>. The comparison is case insensitive. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the following returns TRUE: vpn.host.eq("www.mycompany.com"). If the text mode is URLENCODED the host name is decoded before comparison. For more information, see "Operations for HTTP, HTML, and XML Encoding and "Safe" Characters."
VPN.HOST.SERVER	Extracts the server name part of the host name. For example, if the host name is `www.mycompany.com` or `www.mycompany.com:8080`, the server is `www.mycompany.com`. All text operations after this prefix are case insensitive.

Basic operations on text

September 21, 2020

Basic operations on text include operations for string matching, calculating the length of a string, and controlling case sensitivity. You can include white space in a string that is passed as an argument to an expression, but the string cannot exceed 255 characters.

String comparison functions

The following table lists basic string matching operations in which the functions return a Boolean TRUE or FALSE.

Function	Description
<text>.CONTAINS(<string>)	Returns a Boolean TRUE value if the target contains <string>. Example: `http.req.url.contains(".jpeg")`

Function	Description
<text>.EQ(<string>)	Returns a Boolean TRUE value if the target is an exact match with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.eq("myhostabc")`
<text>.STARTSWITH(<string>)	Returns a Boolean TRUE value if the target begins with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.startswith("myhost")`
<text>.ENDSWITH(<string>)	Returns a Boolean TRUE value if the target ends with <string>. For example, the following expression returns a Boolean TRUE for a URL with a host name of "myhostabc": `http.req.url.hostname.endswith("abc")`
<text>.NE(<string>)	Returns a Boolean TRUE value if the prefix is not equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.GT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.

Function	Description
<text>.GE(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically greater than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.LT(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically lesser than the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.
<text>.LE(<string>)	Returns a Boolean TRUE value if the prefix is alphabetically lesser than or equal to the string argument. If the prefix returns a non-string value, the function argument is compared to the string representation of the value returned by the prefix. You can use the functions with SET_TEXT_MODE(IGNORECASE) or SET_TEXT_MODE(NOIGNORECASE), and with both ASCII and UTF-8 character sets.

Calculate the length of a string

The <text>.LENGTH operation returns a numeric value that is equal to the number of characters (not bytes) in a string:

```
<text>.LENGTH
```

For example, you may want to identify request URLs that exceed a particular length. Following is an expression that implements this example:

```
HTTP.REQ.URL.LENGTH < 500
```

After taking a count of the characters or elements in a string, you can apply numeric operations to them. For more information, see Default Syntax Expressions: Working with Dates, Times, and Numbers.

Consider, ignore, and change text case

The following functions operate on the case (upper-case or lower-case) of the characters in the string.

Function	Description	
<text>.SET_TEXT_MODE(IGNOR	NOIGNORECASE)	This function turns case sensitivity on or off for all text operations.
<text>.TO_LOWER	Converts the target to lowercase for a text block of up to 2 kilobyte (KB). Returns UNDEF if the target exceeds 2 KB. For example, the string "ABCd:" is converted to "abcd:".	
<text>.TO_UPPER	Converts the target to uppercase. Returns UNDEF if the target exceeds 2 KB. For example, the string "abcD:" is converted to "ABCD:".	

Strip specific characters from a string

You can use the STRIP_CHARS(<string>) function to remove specific characters from the text that is returned by a default syntax expression prefix (the input string). All instances of the characters that you specify in the argument are stripped from the input string. You can use any text method on the resulting string, including the methods used for matching the string with a pattern set.

For example, in the expression CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_"), the STRIP_CHARS(<string>) function strips all periods (.), hyphens (-), and underscores (_) from the domain name returned by the prefix CLIENT.UDP.DNS.DOMAIN. If the domain name that is returned is "a.dom_ai_n-name", the function returns the string "adomainname".

In the following example, the resulting string is compared with a pattern set called "listofdomains":

```
CLIENT.UDP.DNS.DOMAIN.STRIP_CHARS(".-_").CONTAINS_ANY("listofdomains")
```

Note: You cannot perform a rewrite on the string that is returned by the STRIP_CHARS(<string>) function.

The following functions strip matching characters from the beginning and end of a given string input.

Function	Description
<text>.STRIP_START_CHARS(s)	Strips matching characters from the beginning of the input string until the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is TestLang and ://*en_us:is its value,HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:/“* the specified characters from the beginning of the value of the header until the first non-matching character e is found and returnsen_us: as a string.
<text>.STRIP_END_CHARS(s)	Strips matching characters from the end of the input string to the first non-matching character is found and returns the remainder of the string. You must specify the characters that you want to strip as a single string within quotation marks. For example, if the name of a header is TestLang and ://*en_us:is its value,HTTP.RES.HEADER(“TestLang”).STRIP_START_CHARS(“:/“* the specified characters from the end of the value of the header until the first non-matching character s is found and returns://_en_us as a string.

Append a string to another string

You can use the APPEND() function to append the string representation of the argument to the string representation of the value returned by the preceding function. The preceding function can be one that returns a number, unsigned long, double, time value, IPv4 address, or IPv6 address. The argument can be a text string, number, unsigned long, double, time value, IPv4 address, or IPv6 address. The resulting string value is the same string value that is obtained by using the + operator.

Complex operations on text

September 21, 2020

In addition to performing simple string matching, you can configure expressions that examine more complex aspects of text, including examining the length of a string and looking within a text block for patterns rather than specific strings.

Be aware of the following for any text-based operation:

- For any operation that takes a string argument, the string cannot exceed 255 characters.
- You can include white space when you specify a string in an expression.

Operations on the length of a string

The following operations extract strings on the basis of a character count.

Character Count Operation	Description
`<text>.TRUNCATE(<count>)`	Returns a string after truncating the end of the target by the number of characters in <count>. If the entire string is shorter than <count>, nothing is returned.
`<text>.TRUNCATE(<character>, <count>)`	Returns a string after truncating the text after <character> by the number of characters specified in <count>.
`<text>.PREFIX(<character>, <count>)`	Selects the longest prefix in the target that has at most <count> occurrences of <character>.
`<text>.SUFFIX(<character>, <count>)`	Selects the longest suffix in the target that has at most `<count>` occurrences of <character>. For example, consider the following response body: JLEwx. The following expression returns a value of "JLEwx": http.res.body(100).suffix('L',1) The following expression returns "LLEwx": http.res.body(100).suffix('L',2)

Character Count Operation	Description
`<text>.SUBSTR(<starting_offset>, <length>)`	Select a string with <length> number of characters from the target object. Begin extracting the string after the `<starting_offset>`. If the number of characters after the offset are fewer than the value of the `<length>` argument, select all the remaining characters.
`<text>.SKIP(<character>, <count>)`	Select a string from the target after skipping over the longest prefix that has at most <count> occurrences of <character>.

Operations on a portion of a string

Refer to the String operations table to know how to extract a subset of a larger string by using one of the operations.

Operations for comparing the alphanumeric order of two strings

The COMPARE operation examines the first nonmatching character of two different strings. This operation is based on lexicographic order, which is the method used when ordering terms in dictionaries.

This operation returns the arithmetic difference between the ASCII values of the first nonmatching characters in the compared strings. The following differences are examples:

- The difference between "abc" and "abd" is -1 (based on the third pair-wise character comparison).
- The difference between "@" and "abc" is -33.
- The difference between "1" and "abc" is -47.

Following is the syntax for the COMPARE operation.

`<text>.COMPARE(<string>)`

Extract an integer from a string of bytes that represent text

Refer to the Integer extraction table to know how to treat a string of bytes that represent text as a sequence of bytes, extract 8, 16, or 32 bits from the sequence, and then convert the extracted bits to an integer.

Convert text to a hash value

You can convert a text string to a hash value by using the HASH function. This function returns a 31-bit positive integer as a result of the operation. Following is the format of the expression:

`<text>.HASH`

This function ignores case and white spaces. For example, after the operation, the two strings Ab c and a bc would produce the same hash value.

Encode and decode text by applying the Base64 encoding algorithm

The following two functions encode and decode a text string by applying the Base64 encoding algorithm

Function	Description
text.B64ENCODE	Encodes the text string (designated by text) by applying the Base64 encoding algorithm.
text.B64DECODE	Decodes the Base64-encoded string (designated by text) by applying the Base64 decoding algorithm. The operation raises an UNDEF if text is not in B64-encoded format.

Refine the search in a rewrite action by using the EXTEND function

The EXTEND function is used in rewrite actions that specify patterns or pattern sets and target the bodies of HTTP packets. When a pattern match is found, the EXTEND function extends the scope of the search by a predefined number of bytes on both sides of the matching string. A regular expression can then be used to perform a rewrite on matches in this extended region. Rewrite actions that are configured with the EXTEND function perform rewrites faster than rewrite actions that evaluate entire HTTP bodies using only regular expressions.

The format of the EXTEND function is EXTEND(m,n), where m and n are the number of bytes by which the scope of the search is extended before and after the matching pattern, respectively. When a match is found, the new search scope comprises m bytes that immediately precede the matching string, the string itself, and the n bytes that follow the string. A regular expression can then be used to perform a rewrite on a portion of this new string.

The EXTEND function can be used only if the rewrite action in which it is used fulfills the following requirements:

- The search is performed by using patterns or patterns sets (not regular expressions)

- The rewrite action evaluates only the bodies of HTTP packets.

Additionally, the EXTEND function can be used only with the following types of rewrite actions:

- replace_all
- insert_after_all
- delete_all
- insert_before_all

For example, you might want to delete all instances of `"<http://exampleurl.com/>"` and `"<http://exampleurl.au/>"` in the first 1000 bytes of the body. To do this, you can configure a rewrite action to search for all instances of the string exampleurl, extend the scope of the search on both sides of the string when a match is found, and then use a regular expression to perform the rewrite in the extended region. The following example extends the scope of the search by 20 bytes to the left and 50 bytes to the right of the matching string:

```
add rewrite action delurl_example delete_all 'HTTP.REQ.BODY(1000)'-pattern
exampleurl -refineSearch 'extend(20,50).regex_select(re##http://exampleurl
.(com|au)##)'
```

Convert text to hexadecimal format

The following function converts text to hexadecimal format and extracts the resulting string:

`<text>.BLOB_TO_HEX(<string>)`

For example, this function converts the byte string “abc” to “61:62:63”.

Encrypt and decrypt text

In default syntax expressions, you can use the ENCRYPT and DECRYPT functions to encrypt and decrypt text. Data encrypted by the ENCRYPT function on a given NetScaler appliance or high availability (HA) pair is intended for decryption by the DECRYPT function on the same NetScaler appliance or HA pair. The appliance supports the RC4, DES3, AES128, AES192, and AES256 encryption methods. The key value that is required for encryption is not user-specifiable. When an encryption method is set, the appliance automatically generates a random key value that is appropriate for the specified method. The default method is AES256 encryption, which is the most secure encryption method and the one that Citrix recommends.

You do not need to configure encryption unless you want to change the encryption method or you want the appliance to generate a new key value for the current encryption method.

Note: You can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see Encrypt and decrypt XML payloads.

Configure encryption

During startup, the appliance runs the set ns encryptionParams command with, by default, the AES256 encryption method, and uses a randomly generated key value that is appropriate for AES256 encryption. The appliance also encrypts the key value and saves the command, with the encrypted key value, to the NetScaler configuration file. Consequently, the AES256 encryption method is enabled for the ENCRYPT and DECRYPT functions by default. The key value that is saved in the configuration file persists across reboots even though the appliance runs the command each time you restart it.

You can run the set ns encryptionParams command manually, or use the GUI, if you want to change the encryption method or if you want the appliance to generate a new key value for the current encryption method. To use the CLI to change the encryption method, set only the method parameter, as shown in "**Example 1: Changing the Encryption Method**." If you want the appliance to generate a new key value for the current encryption method, set the method parameter to the current encryption method and the keyValue parameter to an empty string (""), as shown in "**Example 2: Generating a New Key Value for the Current Encryption Method**." After you generate a new key value, you must save the configuration. If you do not save the configuration, the appliance uses the newly generated key value only until the next restart, after which it reverts to the key value in the saved configuration.

Configure encryption by using the GUI

1. Navigate to **System** > **Settings**.
2. In the **Settings** area, click **Change Encryption** parameters.
3. In the **Change Encryption Parameters** dialog box, do one of the following:
    - To change the encryption method, in the **Method** list, select the encryption method that you want.
    - To generate a new key value for the current encryption method, click **Generate a new key** for the selected method.
4. Click **OK**.

Use the ENCRYPT and DECRYPT functions

You can use the ENCRYPT and DECRYPT functions with any expression prefix that returns text. For example, you can use the ENCRYPT and DECRYPT functions in rewrite policies for cookie encryption. In the following example, the rewrite actions encrypt a cookie named MyCookie, which is set by a back-end service, and decrypt the same cookie when it is returned by a client:

```
add rewrite action my-cookie-encrypt-action replace "HTTP.RES.SET_COOKIE.
COOKIE("MyCookie").VALUE(0)""HTTP.RES.SET_COOKIE.COOKIE("MyCookie").VALUE
(0).ENCRYPT"-bypassSafetyCheck YES
```

```
add rewrite action my-cookie-decrypt-action replace "HTTP.REQ.COOKIE.VALUE(
"MyCookie")""HTTP.REQ.COOKIE.VALUE("MyCookie").DECRYPT"-bypassSafetyCheck
YES
```

After you configure policies for encryption and decryption, save the configuration to bring the policies into effect.

Configure encryption key for third-party encryption

In default syntax expressions, you can use ENCRYPT and DECRYPT functions for encrypting and decrypting text in a request or response. The data encrypted by the ENCRYPT function on an appliance (standalone, high availability or cluster) is intended to be decrypted by the DECRYPT function by the same appliance. The appliance supports RC4, DES, Triple-DES, AES92, and AES256 encryption methods and each of these methods use a secret key for both encryption and decryption of data. You can use the any of these methods to encrypt and decrypt data in two ways - self-encryption and third-party encryption.

The self-encryption feature in an appliance (standalone, high availability or cluster) encrypts and then decrypts data by evaluating the header value. One example to understand this is the HTTP Cookie encryption. The expression evaluates the header, encrypts the HTTP cookie value in the Set-Cookie header in the outgoing response and then decrypts the cookie value when it is returned in the cookie header of a subsequent incoming request from the client. The key value is not user configurable, instead when an encryption method is configured in the set ns encryptionParams command, the appliance automatically generates a random key value for the configured method. By default, the command uses the AES256 encryption method, which is the highly secured method and Citrix recommends this method.

The third-party encryption feature encrypts or decrypts data with a third-party application. For example, a client may encrypt data in a request and the appliance decrypts the data before sending it the back-end server or vice versa. To perform this, the appliance and the third party application must share a secret key. On the appliance, you can directly configure the secret key using an encryption key object and key value is automatically generated by the appliance for a stronger encryption. The same key is manually configured on the third-party appliance so that both appliance and third-party application can use the same key for encrypting and decryption data.

> **Note**:
>
> Using third-party encryption, you can also encrypt and decrypt XML payloads. For information about the functions for encrypting and decrypting XML payloads, see "Encrypting and Decrypting XML Payloads.

Cipher methods

A cipher method provides two functions: an encryption function that transforms a plaintext byte sequence into a ciphertext byte sequence, and a decryption function that transforms the ciphertext back to the plaintext. Cipher methods use byte sequences called keys to perform encryption and decryption. Cipher methods that use the same key for encryption and decryption are called symmetric. Cipher methods that use different keys for encryption and decryption are asymmetric. The most notable examples of asymmetric ciphers are in public key cryptography, which uses a public key available to anyone for encryption and a private key known only to the decryptor.

A good cipher method makes it infeasible to decrypt ("crack") ciphertext if you don't possess the key. "Infeasible" really means that cracking the cyphertext would take more time and computing resources than it is worth. As computers become more powerful and cheaper, ciphers that were formerly infeasible to crack become more feasible. Also, over time, flaws are found in cipher methods (or their implementations), making cracking easier. Newer cipher methods are therefore preferred over older ones. In general, longer length keys provide better security than shorter keys, at the cost of longer encryption and decryption times.

A cipher method can use stream ciphers or block ciphers. RC4 is the mostly secured stream ciphers and it is used only for legacy application. Block ciphers can include padding.

Stream ciphers

A stream cipher method operates on individual bytes. Only one stream cipher is available on NetScaler- appliances: RC4, which uses a 128 bit (16 byte) key length. For a given key, RC4 generates a pseudo-random sequence of bytes, call a keystream, which is X-ORed with the plaintext to produce the ciphertext. RC4 is no longer considered secure and should be used only if required by legacy applications.

Block ciphers

A block cipher method operates on a fixed block of bytes. A NetScaler appliance provides two block ciphers: Data Encryption Standard (DES) and the Advanced Encryption Standard (AES). DES uses a block size of 8 bytes and (on a NetScaler appliance) two choices for key length: 64 bits (8 bytes), of which 56 bits are data and 8 bits are parity, and Triple-DES, a 192 bit (24 byte) key length. AES has a block size of 16 bytes and (on NetScaler) three choices for key length: 128 bits (16 bytes), 192 bits (24 bytes) and 256 bits (32 bytes).

Padding

If the plaintext for a block cipher is not an integral number of blocks, padding with additional bytes might be required. For example, suppose the plaintext is "xyzzy" (hex 78797a7a79). For an 8-byte

Triple-DES block, this value would have to be padded to create 8 bytes. The padding scheme must allow the decryption function to determine the length of the original plaintext after decryption. Following are some padding schemes currently in use (n is the number of bytes added):

- PKCS7: Adds n bytes of value n each. For example, 78797a7a79030303. This is the padding scheme used by OpenSSL and ENCRYPT() policy function. The PKCS5 padding scheme is the same as PKCS7.
- ANSI X.923: Adds n-1 zero bytes and a final byte of value n. For example, 78797a7a79000003.
- ISO 10126: Adds n-1 random bytes and a final byte of value n. For example, 78797a7a79xxxx03, where xx can be any byte value. The DECRYPT() policy function accepts this padding scheme, which also allows it to accept the PKCS7 and ANSI X.923 schemes.
- ISO/IEC 7816-4: Adds a 0x80 byte and n-1 zero bytes. For example, 78797a7a79800000. This is also call OneAndZeros padding.
- Zero: Adds n zero bytes. Example: 78797a7a79000000. This can only be used with plaintext that does not include NUL bytes.

If padding is used and the plaintext is an integral number of blocks, an additional block is typically added so that the decryption function can unambiguously determine the original plaintext length. For PCKS7 and 8 byte block, this would be 0808080808080808.

Modes of operation

There are a number of different modes of operation for block ciphers, which specify how multiple blocks of plaintext are encrypted. Some modes use an initialization vector (IV), a block of data apart from the plaintext that is used to start the encryption process. It is a good practice to use a different IV for each encryption, so that the same plaintext produces different ciphertext. The IV does not need to be secret, and so is usually prepended to the ciphertext. Modes include:

- Electronic Codebook (ECB): Each block of plaintext is encrypted independently. An IV is not used. Padding is required if the plaintext is not a multiple of the cipher block size. The same plaintext and key always produces the same ciphertext. Because of this, ECB is considered less secure than other modes and should only be used for legacy applications.
- Cipher Block Chaining (CBC): Each block of plaintext is XORed with the previous ciphertext block, or the IV for the first block, before being encrypted. Padding is required if the plaintext is not a multiple of the cipher block size. This is the mode used with the NetScaler encryptionParams method.
- Cipher Feedback (CFB): The previous ciphertext block, or the IV for the first block, is encrypted and the output is XORed with the current plaintext block to create the current ciphertext block. The feedback can be 1 bit, 8 bits, or 128 bits. Since the plaintext is XORed with the cipher text, padding is not required.
- Output Feedback (OFB): A keystream is generated by applying the cipher successively to the IV and XORing the keystream blocks with the plaintext. Padding is not required.

Configure encryption keys for third-party encryption

Following are the configuration tasks performed in configuring encryption key.

1. Adding an encryption key. Configures an encryption key for a specified cipher method with a specified key value.
2. Modifying an encryption key. You can edit parameters for a configured encryption key.
3. Unsetting an encryption key. Sets parameters for a configured encryption key to their default values. An encryptionKey value with the name must exist. Sets padding to DEFAULT (determined by the method), Deletes an existing IV, which causes ENCRYPT() to generate a random IV. Deletes an existing comment. The method and key value cannot be reset.
4. Removing an encryption key. Deletes a configured encryption key. The key cannot have any references.
5. Show an encryption key. Displays parameters for the configured encryption key or all configured keys. If the name is omitted, the key value is not displayed.

Add an encryption key by using the CLI

At the command prompt, type:

```
add ns encryptionKey <name> -method <method> [-keyValue <hexstring>][-padding (OFF | ON)] [-iv <hexstring>] [-comment <string>]
```

Where,

```
<method> = ( NONE | RC4 | DES3 | AES128 | AES192 | AES256 | DES | DES-CBC | DES-CFB | DES-OFB | DES-ECB | DES3-CBC | DES3-CFB | DES3-OFB | DES3-ECB | AES128-CBC | AES128-CFB | AES128-OFB | AES128-ECB | AES192-CBC | AES192-CFB | AES192-OFB | AES192-ECB | AES256-CBC | AES256-CFB | AES256-OFB | AES256-ECB ) <hexstring> = hex-encoded byte sequence
<!--NeedCopy-->
```

The above encryption methods specify the operation mode with CBC as the default mode of operation. Therefore, DES, DES2, AES128, AES192, and AES256 methods are equivalent to DES-CBC, DES3-CBC, AES128-CBC, AES192-CBC, and AES256-CBC methods.

Modify an encryption key by using the CLI

At the command prompt, type:

```
set ns encryptionKey <name> [-method <method>] [-keyValue <hexstring>] [-padding ( OFF | ON )] [-iv \<string>] [-comment \<string>]
```

Unset an encryption key by using the CLI

At the command prompt, type:

`unset ns encryptionKey <name> [-padding] [-iv] [-comment]`

Remove an encryption key by using the CLI

At the command prompt, type:

`rm ns encryptionKey <name>`

Show an encryption key by using the CLI

At the command prompt, type:

Example:

```
show ns encryptionKey [<name>]

add ns encryptionKey my_key -method aes256 -keyValue 26
    ea5537b7e0746089476e5658f9327c0b10c3b4778c673a5b38cee182874711  - iv
    c2bf0b2e15c15004d6b14bcdc7e5e365
    set ns encryptionKey my_key -keyValue
        b8742b163abcf62d639837bbee3cef9fb5842d82d00dfe6548831d2bd1d93476
    unset ns encryptionKey my_key -iv
    rm ns encryptionKey my_key
    show ns encryptionKey my_key
    Name: my_key
    Method: AES256
    Padding: DEFAULT
    Key Value: (not disclosed)
<!--NeedCopy-->
```

Add an encryption key by using the GUI

Navigate to **System** > **Encryption Keys** and click **Add** to create an encryption key.

Modify an encryption key by using the GUI

Navigate to **System** > **Encryption Keys** and click **Edit** to modify parameters for a configured encryption key.

Remove an encryption key by using the GUI

Navigate to **System** > **Encryption keys** and click **Delete**.

ENCRYPT and DECRYPT functions for third-party encryption

Following is the ENCRYPT function used for third-part encryption.

`ENCRYPT (encryptionKey, out_encoding)`

Where,

Input data for the appliance is the text to be encrypted

encryptionKey: An optional string parameter that specifies the configured encryption key object to provide the encryption method, secret key value and other encryption parameters. If omitted, the method uses the automatically generated key value associated with the set ns encryptionParamS command.

out_encoding: This value specifies how the output is encoded. If omitted, BASE64 encoding is used.

Input:

```
BASE64: original PEM base64-encoding: 6 bits (0..63) encoded as one
    ASCII character:
                  0..23 = 'A'..'Z', 24..51 = 'a'..'z', 52..61 = '0'..'9
         ', 62 = '+', 63 = '/', '=' = pad byte.
       BASE64URL:  URL and Filename safe base64-encoding: same as BASE64
          except 62 = '-', 63 = '_'
       HEX_UPPER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
         '.
       HEX_LOWER:  Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'a'..'f
         '.
       HEX_COLONS: Hexadecimal with 0..9 = '0'..'9' and 10..15 = 'A'..'F
         '; ':' between each hex byte. Matches BLOB_TO_HEX() output
         format
       HEX:        For input, accepts HEX_UPPER, HEX_LOWER, and
         HEX_COLONS format. For output, produces HEX_LOWER format
<!--NeedCopy-->
```

Output: The output is a text encrypted using the specified method and key and encoded using a specified output encoding. It inserts a generated IV before the encrypted text for block methods and modes that require an IV, and either no IV is specified for the encryptionKey or the encryptionKey is omitted.

Following is the DECRYPT function used for third-part decryption.

`DECRYPT(encryptionKey, in_encoding)`

Where,

Input data is an encrypted text using the specified method and key encoded using the specified input encoding. This text is expected to include a generated IV before the encrypted text for block methods and modes that require an IV, and either no IV is specified for the encryptionKey or the encryptionKey is omitted.

encryptionKey—An optional string parameter that specifies the configured encryptionKey object to provide the encryption method, secret key and other encryption parameters. If omitted, the method and automatically generated key associated with the encryptionParams setting will be used

in_encoding—An optional enumeration parameter that specifies how the input is expected to be encoded. The values are the same as the out_encoding of ENCRYPT. If omitted, BASE64 encoding will be expected.

The output data is an unencoded decrypted text.

Variants and optional parameters

Following are the variants of these functions with the optional parameters:

Variant	Description
ENCRYPT	Use encryptionParams command and BASE64 output encoding parameter.
ENCRYPT(out_encoding)	Use encryptionParams and specified output encoding parameter.
ENCRYPT(encryptionKey)	Use the specified encryptionKey and BASE64 output encoding parameter.
ENCRYPT(encryptionKey, out_encoding)	Use the specified encryptionKey and output encoding parameter.
DECRYPT	Use encryptionParams command and BASE64 input encoding parameter.
DECRYPT(out_encoding)	Use encryptionParams command and the specified input encoding parameter.
DECRYPT(encryptionKey)	Use the specified encryptionKey and BASE64 input encoding parameter.
DECRYPT(encryptionKey, out_encoding)	Use the specified encryptionKey and input encoding parameter.

Configure HMAC keys

NetScaler appliances support a Hashed Message Authentication Code (HMAC) function that calculates a digest method or hash of input text by using a secret key shared between a message sender and message receiver. The digest method (derived from an RFC 2104 technique) authenticates the sender and verifies that the message content has not been altered. For example, when a client sends a message with the shared HMAC key to a NetScaler appliance, advanced (PI) policy expressions use the HMAC function to compute the hash-based code on the selected text. Then, when the receiver receives the message with the secret key, it recomputes the HMAC by comparing it with the original HMAC to determine whether the message has been altered. The HMAC function is supported by standalone appliances and by appliances in a high availability configuration or in a cluster. Using it is similar to configuring an encryption key.

The add ns hmackey and set ns hmackey commands include a parameter that specifies the digest method and the shared secret key to use for the HMAC computation.

To configure a HMAC key, you must perform the following:

1. Adding an HMAC key. Configures an HMAC key with a specified key value.
2. Modifying an HMAC key. Modifies parameters for a configured HMAC key. The digest method can be changed without changing the key value, since the key value length is not determined by the digest. However, it is advisable to specify a new key when changing the digest.
3. Unsetting an HMAC key. Sets parameters for a configured HMAC key to their default values. An hmacKey object with the name must exist. The only parameter that can be unset is the comment, which is deleted.
4. Removing an HMAC key. Deletes a configured key. The key cannot have any references.
5. Show an HMAC key. Displays parameters for the configured HMAC ac key or all configured keys. If the name is omitted, the key value is not displayed.

Add an HMAC key by using the CLI

At the command prompt, type:

```
add ns hmacKey <name> -digest <digest> -keyValue <hexstring> [-comment <
string>]
```

Where,

```
<digest> = (MD2 | MD4 | MD5 | SHA1 | SHA224 | SHA256 | SHA384 | SHA512)
```

This command configures a key for the HMAC () function with the specified digest and key value and also validates the following:

- Name syntax is correct and does not duplicate the name of an existing key.
- Key value is valid hex. As specified by RFC 2104, the key can any length. On a NetScaler appliance, the key value can range from 0 to 255 bytes (0 to 510 hex characters). If the key is shorter than the digest block size, a warning is issued and the key is padded with zeroes. If the key is longer than the block size, it is run through the digest to produce an effective key that is the same length as the block.

Example:

```
add ns hmacKey my_hmac_key -digest sha1 -keyValue 0c753c6c5ef859189cacdf95b506d02c179
```

The above encryption methods specify the operation mode with CBC as the default mode of operation. Therefore, DES, DES2, AES128, AES192, and AES256 methods are equivalent to DES-CBC, DES3-CBC, AES128-CBC, AES192-CBC, and AES256-CBC methods.

Modify an HMAC key by using the CLI

This command modifies the parameters configured for a HMAC key. You can change the digest without changing the key value, since the key value length is not determined by the digest. However, it is advisable to specify a new key when changing the digest. At the command prompt, type:

```
set ns hmacKey <name> [-digest <digest>] [-keyValue <hexvalue>] [-comment <
string>]
```

Unset an HMAC key by using the CLI

This command sets parameters configured for a HMAC key with their default values. An hmacKey object with the name must exist. The only parameter that you can unset is the comment option, which is deleted. At the command prompt, type:

```
unset ns hmacKey <name> -comment
```

Remove an HMAC key by using the CLI

This command deletes the configured hmac key. The key cannot have references. At the command prompt, type:

```
rm ns hmacKey <name>
```

Show an HMAC key by using the CLI

At the command prompt, type:

```
show ns encryptionKey [<name>]

add ns hmacKey my_hmac_key -digest sha1 -keyValue 0
    c753c6c5ef859189cacdf95b506d02c1797407d
set ns hmacKey my_hmac_key -keyValue
    f348c594341a840a1f641a1cf24aa24c15eb1317
rm ns hmacKey my_hmac_key
```

```
show ns hmacKey my_hmac_key
          Name: my_hmac_key
    Digest: SHA1
    Key Value: (not disclosed)
<!--NeedCopy-->
```

Advanced policy expressions: working with dates, times, and numbers

September 21, 2020

Most numeric data that the NetScaler appliance processes consists of dates and times. In addition to working with dates and times, the appliance processes other numeric data, such as the lengths of HTTP requests and responses. To process this data, you can configure advanced policy expressions that process numbers.

A numeric expression consists of an expression prefix that returns a number and sometimes, but not always, an operator that can perform an operation on the number. Examples of expression prefixes that return numbers are SYS.TIME.DAY, HTTP.REQ.CONTENT_LENGTH, and HTTP.RES.BODY.LENGTH. Numeric operators can work with any prefix expression that returns data in numeric format. The GT(<int>) operator, for example, can be used with any prefix expression, such as HTTP.REQ.CONTENT_LENGTH, that returns an integer. Numeric expression prefixes and operators are also covered in Compound Operations for Numbers and Advanced Policy Expressions: Parsing HTTP, TCP, and UDP Data.

Format of dates and times in an expression

September 21, 2020

When configuring an advanced policy expression in a policy that works with dates and times (for example, the NetScaler system time or a date in an SSL certificate), you specify a time format as follows:

`GMT|LOCAL [<yyyy>] [<month>] [<d>] [<h>] [<m>] [<s>]`

Where:

- <yyyy> is a four-digit year after GMT or LOCAL.
- <month> is a three-character abbreviation for the month, for example, Jan, Dec.
- <d> is a day of the week or an integer for the date.

You cannot specify the day as Monday, Tuesday, and so on. You specify either an integer for a specific day of the month, or you specify a date as the first, second, third weekday of the month, and so on. Following are examples of specifying a day of the week:

  - Sun_1 is the first Sunday of the month.
  - Sun_3 is the third Sunday of the month.
  - Wed_3 is the third Wednesday of the month.
  - 30 is an example of an exact date in a month.

- <h> is the hour, for example, 10h.
- <s> is the number of seconds, for example, 30s.

The following example expression is true if the date is between 2008 Jan and 2009 Jan, based on GMT.

`http.req.date.between(GMT 2008 Jan, GMT 2009 Jan)`

The following example expression is true for March and all months that follow March in the calendar year, based on GMT:

`sys.time.ge(GMT 2008 Mar)`

When you specify a date and time, note that the format is case sensitive and must preserve the exact number of blank spaces between entries.

```
**Note:**

In an expression that requires two time values, both must use GMT or both must use LOCAL. You cannot mix the two in an expression.

Unlike when you use the SYS.TIME prefix in an advanced policy expression, if you specify SYS.TIME in a rewrite action, the NetScaler returns a string in conventional date format (for example, Sun, 06 Nov 1994 08:49:37 GMT). For example, the following rewrite action replaces the http.res.date header with the NetScaler system time in a conventional date format:

add rewrite action sync_date replace http.res.date sys.time
```

Expressions for the NetScaler system time

September 21, 2020

The SYS.TIME expression prefix extracts the NetScaler system time. You can configure expressions that establish whether a particular event occurred at a particular time or within a particular time range according to the NetScaler system time.

The following table describes the expressions that you can create by using the SYS.TIME prefix.

- **SYS.TIME.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE if the returned value is later than <time1> and earlier than <time2>.

  You format the <time1>, <time2> arguments as follows:

  - They must both be GMT or both LOCAL.
  - <time2> must be later than <time1>.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following:

  - sys.time.between(GMT 2004, GMT 2006)
  - sys.time.between(GMT 2004 Jan, GMT 2006 Nov)
  - sys.time.between(GMT 2004 Jan, GMT 2006)
  - sys.time.between(GMT 2005 May Sun_1, GMT 2005 May Sun_3)
  - sys.time.between(GMT 2005 May 1, GMT May 2005 1)
  - sys.time.between(LOCAL 2005 May 1, LOCAL May 2005 1)

- **SYS.TIME.DAY:**

  Returns the current day of the month as a number from 1 through 31.

- **SYS.TIME.EQ(<time>):**

  Returns a Boolean TRUE if the current time is equal to the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.eq(GMT 2005) (TRUE in this example.)
  - sys.time.eq(GMT 2005 Dec) (FALSE in this example.)
  - sys.time.eq(LOCAL 2005 May) (Evaluates to TRUE or FALSE in this example, depending on the current time zone.)
  - sys.time.eq(GMT 10h) (TRUE in this example.)
  - sys.time.eq(GMT 10h 30s) (TRUE in this example.)
  - sys.time.eq(GMT May 10h) (TRUE in this example.)
  - sys.time.eq(GMT Sun) (TRUE in this example.)
  - sys.time.eq(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.NE(<time>):**

  Returns a Boolean TRUE if the current time is not equal to the <time> argument.

- **SYS.TIME.GE(<time>):**

  Returns a Boolean TRUE if the current time is later than or equal to <time>.

For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.ge(GMT 2004) (TRUE in this example.)
  - sys.time.ge(GMT 2005 Jan) (TRUE in this example.)
  - sys.time.ge(LOCAL 2005 May) (TRUE or FALSE in this example, depending on the current time zone.)
  - sys.time.ge(GMT 8h) (TRUE in this example.)
  - sys.time.ge(GMT 30m) (FALSE in this example.)
  - sys.time.ge(GMT May 10h) (TRUE in this example.)
  - sys.time.ge(GMT May 10h 0m) (TRUE in this example.)
  - sys.time.ge(GMT Sun) (TRUE in this example.)
  - sys.time.ge(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.GT(<time>):**

  Returns a Boolean TRUE if the time value is later than the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.gt(GMT 2004) (TRUE in this example.)
  - sys.time.gt(GMT 2005 Jan) (TRUE in this example.)
  - sys.time.gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - sys.time.gt(GMT 8h) (TRUE in this example.)
  - sys.time.gt(GMT 30m) (FALSE in this example.)
  - sys.time.gt(GMT May 10h) (FALSE in this example.)
  - sys.time.gt(GMT May 10h 0m) (TRUE in this example.)
  - sys.time.gt(GMT Sun) (FALSE in this example.)
  - sys.time.gt(GMT May Sun_1) (FALSE in this example.)

- **SYS.TIME.HOURS:**

  Returns the current hour as an integer from 0 to 23.

- **SYS.TIME.LE(<time>):**

  Returns a Boolean TRUE if the current time value precedes or is equal to the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.le(GMT 2006) (TRUE in this example.)
  - sys.time.le(GMT 2005 Dec) (TRUE in this example.)
  - sys.time.le(LOCAL 2005 May) (TRUE or FALSE depending on the current timezone.)
  - sys.time.le(GMT 8h) (FALSE in this example.)
  - sys.time.le(GMT 30m) (TRUE in this example.)

  - sys.time.le(GMT May 10h) (TRUE in this example.)
  - sys.time.le(GMT Jun 11h) (TRUE in this example.)
  - sys.time.le(GMT Wed) (TRUE in this example.)
  - sys.time.le(GMT May Sun_1) (TRUE in this example.)

- **SYS.TIME.LT(<time>):**

  Returns a Boolean TRUE if the current time value precedes the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.lt(GMT 2006) (TRUE in this example.)
  - sys.time.lt.time.lt(GMT 2005 Dec) (TRUE in this example.)
  - sys.time.lt(LOCAL 2005 May) (TRUE or FALSE depending on the current time zone.)
  - sys.time.lt(GMT 8h) (FALSE in this example.)
  - sys.time.lt(GMT 30m) (TRUE in this example.)
  - sys.time.lt(GMT May 10h) (FALSE in this example.)
  - sys.time.lt(GMT Jun 11h) (TRUE in this example.)
  - sys.time.lt(GMT Wed) (TRUE in this example.)
  - sys.time.lt(GMT May Sun_1) (FALSE in this example.)

- **SYS.TIME.MINUTES:**

  Returns the current minute as an integer from 0 to 59.

- **SYS.TIME.MONTH:**

  Extracts the current month and returns an integer from 1 (January) to 12 (December).

- **SYS.TIME.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled reboot, and returns an integer.

  If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **SYS.TIME.RELATIVE_NOW:**

  Calculates the number of seconds between the current NetScaler system time and the specified time, and returns an integer showing the difference.

  If the designated time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **SYS.TIME.SECONDS:**

  Extracts the seconds from the current NetScaler system time, and returns that value as an integer from 0 to 59.

- **SYS.TIME.WEEKDAY:**

  Returns the current weekday as a value from 0 (Sunday) to 6 (Saturday).

- **SYS.TIME.WITHIN (<time1>, <time2>):**

  If you omit an element of time in <time1>, for example, the day or hour, it is assumed to have the lowest value in its range. If you omit an element in <time2>, it is assumed to have the highest value of its range.

  The ranges for the elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. If you specify the year, you must do so in both <time1> and <time2>.

  For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are shown in parentheses):

  - sys.time.within(GMT 2004, GMT 2006) (TRUE in this example.)
  - sys.time.within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - sys.time.within(GMT Feb, GMT) (TRUE, May is in the range of February to December.)
  - sys.time.within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.)
  - sys.time.within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE in this example.)
  - sys.time.within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)

- **SYS.TIME.YEAR:**

  Extracts the year from the current system time and returns that value as a four-digit integer.

Expressions for SSL certificate dates

September 21, 2020

You can determine the validity period for SSL certificates by configuring an expression that contains the following prefix:

```
CLIENT.SSL.CLIENT_CERT
```

The following example expression matches a particular time for expiration with the information in the certificate:

```
client.ssl.client_cert.valid_not_after.eq(GMT 2009)
```

The following table describes time-based operations on SSL certificates. To obtain the expression you want, replace *certificate* in the expression in the first column with the prefix expression, "CLIENT.SSL.CLIENT_CERT".

- **<certificate>.VALID_NOT_AFTER:**

  Returns the last day before certificate expiration. The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).

- **<certificate>.VALID_NOT_AFTER.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE value if the certificate validity is between the <time1> and <time2> arguments. Both <time1> and <time2> must be fully specified. Following are examples:

  GMT 1995 Jan is fully specified.

  GMT Jan is not fully specified

  GMT 1995 20 is not fully specified.

  GMT Jan Mon_2 is not fully specified.

  The <time1> and <time2> arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>.

  For example, if it is GMT 2005 May 1 10h 15m 30s, and the first Sunday of the month, you can specify the following (evaluation results are in parentheses).

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)

- **<certificate>.VALID_NOT_AFTER.DAY:**

  Extracts the last day of the month that the certificate is valid, and returns a number from 1 through 31, as appropriate for the date.

- **<certificate>.VALID_NOT_AFTER.EQ(<time>):**

  Returns a Boolean TRUE if the time is equal to the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following (evaluation results for this example are in parentheses):

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone)

  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)
  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GE(<time>):**

  Returns a Boolean TRUE if the time value is greater than or equal to the argument <time>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.GT(<time>):**

  Returns a Boolean TRUE if the time value is greater than the argument <time>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT Sun) (FALSE)
  - . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.HOURS:**

  Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.

- **<certificate>.VALID_NOT_AFTER.LE(<time>):**

Returns a Boolean TRUE if the time precedes or is equal to the <time> argument.

For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .le(GMT 8h) (FALSE)
  - . . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_AFTER.LT(<time>):**

  Returns a Boolean TRUE if the time precedes the <time> argument.

  For example, if the current time is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month, you can specify the following:

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_AFTER.MINUTES:**

  Extracts the last minute that the certificate is valid and returns that value as an integer from 0 to 59.

- **<certificate>.VALID_NOT_AFTER.MONTH:**

  Extracts the last month that the certificate is valid and returns that value as an integer from 1 (January) to 12 (December).

- **<certificate>.VALID_NOT_AFTER.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled reboot and returns an integer. If the closest boot time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_AFTER.RELATIVE_NOW;**

  Calculates the number of seconds between the current system time and the specified time and returns an integer. If the time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_AFTER.SECONDS:**

  Extracts the last second that the certificate is valid and returns that value as an integer from 0 to 59.

- **<certificate>.VALID_NOT_AFTER.WEEKDAY:**

  Extracts the last weekday that the certificate is valid. Returns a number between 0 (Sunday) and 6 (Saturday) to give the weekday in the time value.

- **<certificate>.VALID_NOT_AFTER.WITHIN(<time1>, <time2>):**

  Returns a Boolean TRUE if the time lies within all the ranges defined by the elements in <time1> and <time2>.

  If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element from <time2>, it is assumed to have the highest value of its range. If you specify a year in <time1>, you must specify it in <time2>.

  The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59. For the result to be TRUE, each element in the time must exist in the corresponding range that you specify in <time1>, <time2>.

  For example, if time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses):

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - . . .within(GMT Feb, GMT) (TRUE, May is in the range for February to December)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday lies within the range of the first Sunday through the third Sunday)
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone)

- **<certificate>.VALID_NOT_AFTER.YEAR:**

  Extracts the last year that the certificate is valid and returns a four-digit integer.

- **<certificate>.VALID_NOT_BEFORE:**

  Returns the date that the client certificate becomes valid.

The return format is the number of seconds since GMT January 1, 1970 (0 hours, 0 minutes, 0 seconds).

- **<certificate>.VALID_NOT_BEFORE.BETWEEN(<time1>, <time2>):**

  Returns a Boolean TRUE if the time value is between the two time arguments. Both <time1> and <time2> arguments must be fully specified.

  Following are examples:

  GMT 1995 Jan is fully specified.
  GMT Jan is not fully specified.
  GMT 1995 20 is not fully specified.
  GMT Jan Mon_2 is not fully specified.
  The time arguments must be both GMT or both LOCAL, and <time2> must be greater than <time1>.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .between(GMT 2004, GMT 2006) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006 Nov) (TRUE)
  - . . .between(GMT 2004 Jan, GMT 2006) (TRUE)
  - . . .between(GMT 2005 May Sun_1, GMT 2005 May Sun_3) (TRUE)
  - . . .between(GMT 2005 May 1, GMT May 2005 1) (TRUE)
  - . . .between(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone.)

- **<certificate>.VALID_NOT_BEFORE.DAY:**

  Extracts the last day of the month that the certificate is valid and returns that value as a number from 1 through 31 representing that day.

- **<certificate>.VALID_NOT_BEFORE.EQ(<time>):**

  Returns a Boolean TRUE if the time is equal to the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .eq(GMT 2005) (TRUE)
  - . . .eq(GMT 2005 Dec) (FALSE)
  - . . .eq(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .eq(GMT 10h) (TRUE)
  - . . .eq(GMT 10h 30s) (TRUE)

  - . . .eq(GMT May 10h) (TRUE)
  - . . .eq(GMT Sun) (TRUE)
  - . . .eq(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GE(<time>):**

  Returns a Boolean TRUE if the time is greater than (after) or equal to the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses):

  - . . .ge(GMT 2004) (TRUE)
  - . . .ge(GMT 2005 Jan) (TRUE)
  - . . .ge(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .ge(GMT 8h) (TRUE)
  - . . .ge(GMT 30m) (FALSE)
  - . . .ge(GMT May 10h) (TRUE)
  - . . .ge(GMT May 10h 0m) (TRUE)
  - . . .ge(GMT Sun) (TRUE)
  - . . .ge(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.GT(<time>):**

  Returns a Boolean TRUE if the time occurs after the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results are in parentheses):

  - . . .gt(GMT 2004) (TRUE)
  - . . .gt(GMT 2005 Jan) (TRUE)
  - . . .gt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .gt(GMT 8h) (TRUE)
  - . . .gt(GMT 30m) (FALSE)
  - . . .gt(GMT May 10h) (FALSE)
  - . . .gt(GMT May 10h 0m) (TRUE)
  - . . .gt(GMT Sun) (FALSE)
  - . . .gt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.HOURS:**

  Extracts the last hour that the certificate is valid and returns that value as an integer from 0 to 23.

- **<certificate>.VALID_NOT_BEFORE.LE(<time>)

  Returns a Boolean TRUE if the time precedes or is equal to the <time> argument.

For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .le(GMT 2006) (TRUE)
  - . . .le(GMT 2005 Dec) (TRUE)
  - . . .le(LOCAL 2005 May) (TRUE or FALSE,depending on the current time zone.)
  - . . .le(GMT 8h) (FALSE)

    . - . .le(GMT 30m) (TRUE)
  - . . .le(GMT May 10h) (TRUE)
  - . . .le(GMT Jun 11h) (TRUE)
  - . . .le(GMT Wed) (TRUE)
  - . . .le(GMT May Sun_1) (TRUE)

- **<certificate>.VALID_NOT_BEFORE.LT(<time>):**

  Returns a Boolean TRUE if the time precedes the <time> argument.

  For example, if the time value is GMT 2005 May 1 10h 15m 30s, and it is the first Sunday of the month of May in 2005, you can specify the following (evaluation results for this example are in parentheses):

  - . . .lt(GMT 2006) (TRUE)
  - . . .lt(GMT 2005 Dec) (TRUE)
  - . . .lt(LOCAL 2005 May) (TRUE or FALSE, depending on the current time zone.)
  - . . .lt(GMT 8h) (FALSE)
  - . . .lt(GMT 30m) (TRUE)
  - . . .lt(GMT May 10h) (FALSE)
  - . . .lt(GMT Jun 11h) (TRUE)
  - . . .lt(GMT Wed) (TRUE)
  - . . .lt(GMT May Sun_1) (FALSE)

- **<certificate>.VALID_NOT_BEFORE.MINUTES:**

  Extracts the last minute that the certificate is valid. Returns the current minute as an integer from 0 to 59.

- **<certificate>.VALID_NOT_BEFORE.MONTH:**

  Extracts the last month that the certificate is valid. Returns the current month as an integer from 1 (January) to 12 (December).

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_BOOT:**

  Calculates the number of seconds to the closest previous or scheduled NetScaler reboot and returns an integer. If the closest boot time is in the past, the integer is negative; if it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_BEFORE.RELATIVE_NOW:**

  Returns the number of seconds between the current NetScaler system time and the specified time as an integer. If the designated time is in the past, the integer is negative. If it is in the future, the integer is positive.

- **<certificate>.VALID_NOT_BEFORE.SECONDS:**

  Extracts the last second that the certificate is valid. Returns the current second as an integer from 0 to 59.

- **<certificate>.VALID_NOT_BEFORE.WEEKDAY:**

  Extracts the last weekday that the certificate is valid. Returns the weekday as a number between 0 (Sunday) and 6 (Saturday).

- **<certificate>.VALID_NOT_BEFORE.WITHIN(<time1>, <time2>):**

  Returns a Boolean TRUE if each element of time exists within the range defined in the <time1>, <time2> arguments.

  If you omit an element of time from <time1>, it is assumed to have the lowest value in its range. If you omit an element of time from <time2>, it is assumed to have the highest value in its range. If you specify a year in <time1>, it must be specified in <time2>. The ranges for elements of time are as follows: month 1-12, day 1-31, weekday 0-6, hour 0-23, minutes 0-59 and seconds 0-59.

  For example, if the time is GMT 2005 May 10 10h 15m 30s, and it is the second Tuesday of the month, you can specify the following (evaluation results are in parentheses):

  - . . .within(GMT 2004, GMT 2006) (TRUE)
  - . . .within(GMT 2004 Jan, GMT 2006 Mar) (FALSE, May is not in the range of January to March.)
  - . . .within(GMT Feb, GMT) (TRUE, May is in the range of February to December.)
  - . . .within(GMT Sun_1, GMT Sun_3) (TRUE, the second Tuesday is between the first Sunday and the third Sunday.)
  - . . .within(GMT 2005 May 1 10h, GMT May 2005 1 17h) (TRUE)
  - . . .within(LOCAL 2005 May 1, LOCAL May 2005 1) (TRUE or FALSE, depending on the NetScaler system time zone)

- **<certificate>.VALID_NOT_BEFORE.YEAR:**

  Extracts the last year that the certificate is valid. Returns the current year as a four-digit integer.

Expressions for HTTP request and response dates

September 21, 2020

The following expression prefixes return the contents of the HTTP Date header as text or as a date object. These values can be evaluated as follows:

- As a number. The numeric value of an HTTP Date header is returned in the form of the number of seconds since Jan 1, 1970.

  For example, the expression http.req.date.mod(86400) returns the number of seconds since the beginning of the day. These values can be evaluated using the same operations as other non-date-related numeric data. For more information, see Expression prefixes for numeric data other than date and time.

- As an HTTP header. Date headers can be evaluated using the same operations as other HTTP headers.

  For more information, see Default syntax expressions: Parsing HTTP, TCP, and UDP data.

- As text. Date headers can be evaluated using the same operations as other strings.

For more information, see Advanced Policy Expressions: Evaluating Text.

Prefix	Description
HTTP.REQ.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 08:49:37 GMT, RFC850. Sunday, 06-Jan-80 09:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.
HTTP.RES.DATE	Returns the contents of the HTTP Date header as text or as a date object. The date formats recognized are: RFC822. Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, and ASCTIME. Sun Jan 6 08:49:37 1980.

Generate the day of the week, as a string, in short and long formats

September 21, 2020

The functions, WEEKDAY_STRING_SHORT and WEEKDAY_STRING, generate the day of the week, as a string, in short and long formats, respectively. The strings that are returned are always in English. The prefix used with these functions must return the day of the week in integer format and the acceptable range for the value returned by the prefix is 0-6. Therefore, you can use any prefix that returns an integer in the acceptable range. An UNDEF condition is raised if the returned value is not in this range or if memory allocation fails.

Following are the descriptions of the functions:

Function	Description
<prefix>.WEEKDAY_STRING_SHORT	Returns the day of the week in short format. The short form is always 3 characters long with an initial capital and the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING_SHORT returns Sun if the value returned by the WEEKDAY function is 0 and Sat if the value returned by the prefix is 6.
<prefix>.WEEKDAY_STRING	Returns the day of the week in long format. The long form always has an initial capital, with the remaining characters in lower case. For example, SYS.TIME.WEEKDAY.WEEKDAY_STRING returns Sunday if the value returned by the WEEKDAY function is 0 and Saturday if the value returned by the prefix is 6.

Expression prefixes for numeric data other than date and time

September 21, 2020

In addition to configuring expressions that operate on time, you can configure expressions for the following types of numeric data:

- The length of HTTP requests, the number of HTTP headers in a request, and so on.

  For more information, see Expressions for numeric HTTP payload data other than dates.

- IP and MAC addresses.

  For more information, see Expressions for IP addresses and IP subnets.

- Client and server data in regard to interface IDs and transaction throughput rate.

  For more information, see Expressions for numeric client and server data.

- Numeric data in client certificates other than dates.

  For information on these prefixes, including the number of days until certificate expiration and the encryption key size, see Prefixes for numeric data in SSL certificates.

Converting numbers to text

September 21, 2020

The following functions produce binary strings from a number returned by an expression prefix. These functions are particularly useful in the TCP rewrite feature as replacement strings for binary data. For more information about the TCP rewrite feature, see Rewrite.

All the functions return a value of type text. The endianness that some of the functions accept as a parameter is either LITTLE_ENDIAN or BIG_ENDIAN.

Function	Description
<number>.SIGNED8_STRING	Produces an 8-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_SIGNED8(16).SUB(3).SIGNED8_
<number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED8(31).ADD(3).UNSIG
<number>.SIGNED16_STRING(<endianness>)	Produces a 16-bit signed binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).SKIP(12).GET_SIGNED16(0, BIG_ENDIAN).SUB(4).SIGNED16_STRING(BIG_ENDIAN)
<number>.UNSIGNED16_STRING(<endianness>)	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED16(47, LITTLE_ENDIAN).ADD(7).UNSIGNED16_STRING(LITTLE_ENDIAN)

Function	Description
<number>.SIGNED32_STRING(<endianness>)	Produces a 32-bit signed binary string representing the number. Example: HTTP.REQ.BODY(100).AFTER_STR("delim").GET_SIGNED3 BIG_ENDIAN).SUB(1).SIGNED32_STRING(BIG_ENDIAN)
<unsigned_long_number>.UNSIGNED8_STRING	Produces an 8-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED8(24).TYPECAST_UN
<unsigned_long_number>.UNSIGNED16_STRING	Produces a 16-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).GET_UNSIGNED16(23, LITTLE_ENDIAN).TYPECAST_UNSIGNED_LONG_AT.ADD(10).UNSIG
<unsigned_long_number>.UNSIGNED32_STRING(<endianness>)	Produces a 32-bit unsigned binary string representing the number. If the value is out of range, an undef condition is raised. Example: HTTP.REQ.BODY(100).AFTER_STR("delim2").GET_UNSIGN BIG_ENDIAN).ADD(2).UNSIGNED32_STRING(BIG_ENDIAN)

Virtual server based expressions

September 21, 2020

The
SYS.VSERVER("<vserver-name>") expression prefix enables you to identify a virtual server. You can use the following functions with this prefix to retrieve information related to the specified virtual server:

- **THROUGHPUT.** Returns the throughput of the virtual server in Mbps (Megabits per second). The value returned is an unsigned long number.

  Usage: SYS.VSERVER("vserver").THROUGHPUT

- **CONNECTIONS.** Returns the number of connections being managed by the virtual server. The value returned is an unsigned long number.

  Usage: SYS.VSERVER("vserver").CONNECTIONS

- **STATE.** Returns the state of the virtual server. The value returned is UP, DOWN, or OUT_OF_SERVICE. One of these values can therefore be passed as an argument to the EQ() operator to perform a comparison that results in a Boolean TRUE or FALSE.

  Usage: SYS.VSERVER("vserver").STATE

- **HEALTH.** Returns the percentage of services in an UP state for the specified virtual server. The value returned is an integer.

  Usage: SYS.VSERVER("vserver").HEALTH

- **RESPTIME.** Returns the response time as an integer representing the number of microseconds. Response time is the average TTFB (Time To First Byte) from all the services bound to the virtual server.

  Usage: SYS.VSERVER("vserver").RESPTIME

- **SURGECOUNT.** Returns the number of requests in the surge queue of the virtual server. The value returned is an integer.

  Usage: SYS.VSERVER("vserver").SURGECOUNT

Example 1:

The following rewrite policy aborts rewrite processing if the number of connections at the load balancing virtual server LBvserver exceeds 10000:

```
add rewrite policy norewrite_pol sys.vserver("LBvserver").connections.gt
(10000)norewrite
```

Example 2:

The following rewrite action inserts a custom header, TP, whose value is the throughout at the virtual server LBvserver:

```
add rewrite action tp_header insert_http_header TP SYS.VSERVER("LBvserver")
.THROUGHPUT
```

Example 3:

The following audit log message action writes the average TTFB of the services bound to a virtual server, to the newnslog log file:

```
add audit messageaction log_vserver_resptime_act INFORMATIONAL "\"NS
Response Time to Servers:\" + sys.vserver(\"ssllb\").resptime + \" millisec
\""-logtoNewnslog YES -bypassSafetyCheck YES
```

Advanced policy expressions: Parsing HTTP, TCP, and UDP data

September 21, 2020

You can configure Advanced policy expressions to evaluate and process the payload in HTTP requests and responses. The payload associated with an HTTP connection includes the various HTTP headers (both standard and custom headers), the body, and other connection information such as the URL. Additionally, you can evaluate and process the payload in a TCP or UDP packet. For HTTP connections, for example, you can check whether a particular HTTP header is present or if the URL includes a particular query parameter.

You can configure expressions to transform the URL encoding and apply HTML or XML "safe" coding for subsequent evaluation. You can also use XPATH and JSON prefixes to evaluate date in XML and JSON files, respectively.

You can also use text-based and numeric Advanced policy expressions to evaluate HTTP request and response data. For more information, see Advanced policy expressions: Evaluating text. and Default Syntax Expressions: Working with Dates, Times, and Numbers.

About evaluating HTTP and TCP payload

September 21, 2020

The payload of an HTTP request or response consists of HTTP protocol information such as headers, a URL, body content, and version and status information. When you configure a default syntax expression to evaluate HTTP payload, you use a default syntax expression prefix and, if necessary, an operator.

For example, you use the following expression, which includes the http.req.header("<header_name>") prefix and the exists operator, if you want to determine whether an HTTP connection includes a custom header named "myHeader":

```
http.req.header("myHeader").exists
```

You can also combine multiple Advanced policy expressions with Boolean and arithmetic operators. For example, the following compound expression could be useful with various NetScaler features, such as Integrated Caching, Rewrite, and Responder. This expression first uses the && Boolean operator to determine whether an HTTP connection includes the Content-Type header with a value of "text/html." If that operation returns a value of FALSE, the expression determines whether the HTTP connection includes a "Transfer-Encoding" or "Content-Length" header.

```
(http.req.header("Content-Type").exists && http.req.header("Content-Type")
```

`.eq("text/html"))|| (http.req.header("Transfer-Encoding").exists)|| (http.req.header("Content-Length").exists)`

The payload of a TCP or UDP packet is the data portion of the packet. You can configure Advanced policy expressions to examine features of a TCP or UDP packet, including the following:

- Source and destination domains
- Source and destination ports
- The text in the payload
- Record types

The following expression prefixes extract text from the body of the payload:

- HTTP.REQ.BODY(integer). Returns the body of an HTTP request as a multiline text object, up to the character position designated in the integer argument. If there are fewer characters in the body than is specified in the argument, the entire body is returned.
- HTTP.RES.BODY(integer). Returns a portion of the HTTP response body. The length of the returned text is equal to the number in the integer argument. If there are fewer characters in the body than is specified in integer, the entire body is returned.
- CLIENT.TCP.PAYLOAD(integer). Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the integer argument.

Following is an example that evaluates to TRUE if a response body of 1024 bytes contains the string "https", and this string occurs after the string "start string" and before the string "end string":

`http.res.body(1024).after_str("start_string").before_str("end_string").contains("https")`

Note: You can apply any text operation to the payload body. For information on operations that you can apply to text, see
Advanced policy expressions: Evaluating text.

Expressions for identifying the protocol in an incoming IP packet

September 21, 2020

The following table lists the expressions that you can use to identify the protocol in an incoming packet.

Expression	Description
CLIENT.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by clients.

Expression	Description
CLIENT.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by clients.
SERVER.IP.PROTOCOL	Identifies the protocol in IPv4 packets sent by servers.
SERVER.IPV6.PROTOCOL	Identifies the protocol in IPv6 packets sent by servers.

Arguments to the PROTOCOL function

You can pass the Internet Assigned Numbers Authority (IANA) protocol number to the PROTOCOL function. For example, if you want to determine whether the protocol in an incoming packet is TCP, you can use CLIENT.IP.PROTOCOL.EQ(6), where 6 is the IANA-assigned protocol number for TCP. For some protocols, you can pass an enumeration value instead of the protocol number. For example, instead of CLIENT.IP.PROTOCOL.EQ(6), you can use CLIENT.IP.PROTOCOL.EQ(TCP). The following table lists the protocols for which you can use enumeration values, and the corresponding enumeration values for use with the PROTOCOL function.

Protocol	Enumeration value
Transmission Control Protocol (TCP)	TCP
User Datagram Protocol (UDP)	UDP
Internet Control Message Protocol (ICMP)	ICMP
IP Authentication Header (AH), for providing authentication services in IPv4 and IPv6	AH
Encapsulating Security Payload (ESP) protocol	ESP
General Routing Encapsulation (GRE)	GRE
IP-within-IP Encapsulation Protocol	IPIP
Internet Control Message Protocol for IPv6 (ICMPv6)	ICMPv6
Fragment Header for IPv6	FRAGMENT

Use case scenarios

The protocol expressions can be used in both request-based and response-based policies. You can use the expressions in various NetScaler features, such as load balancing, WAN optimization, content switching, rewrite, and listen policies. You can use the expressions with functions such as EQ() and NE(), to identify the protocol in a policy and perform an action.

Following are some use cases for the expressions:

- In Branch Repeater load balancing configurations, you can use the expressions in a listen policy for the wildcard virtual server. For example, you can configure the wildcard virtual server with the listen policy CLIENT.IP.PROTOCOL.EQ(TCP) so that the virtual server processes only TCP traffic and simply bridges all non-TCP traffic. Even though you can use an Access Control List instead of the listen policy, the listen policy provides better control over what traffic is processed.
- For content switching virtual servers of type ANY, you can configure content switching policies that switch requests on the basis of the protocol in incoming packets. For example, you can configure content switching policies to direct all TCP traffic to one load balancing virtual server and all non-TCP traffic to another load balancing virtual server.
- You can use the client-based expressions to configure persistence based on the protocol. For example, you can use CLIENT.IP.PROTOCOL to configure persistence on the basis of the protocols in incoming IPv4 packets.

Expressions for HTTP and cache-control headers

September 21, 2020

One common method of evaluating HTTP traffic is to examine the headers in a request or a response. A header can perform a number of functions, including the following:

- Provide cookies that contain data about the sender.
- Identify the type of data that is being transmitted.
- Identify the route that the data has traveled (the Via header).

> **Note**
>
> If an operation is used to evaluate both header and text data, the header-based operation always overrides the text-based operation. For example, the AFTER_STR operation, when applied to a header, overrides text-based AFTER_STR operations for all instances of the current header type.

Prefixes for HTTP headers

The following table describes expression prefixes that extract HTTP headers.

HTTP Header Prefix	Description
HTTP.REQ.HEADER(“”)	Returns the contents of the HTTP header specified by the argument. The header name cannot exceed 32 characters. Note that this prefix returns the value from the Host header by default. To use this value as a hostname you need to typecast it as follows: http.req.header(” more information on typecasting, see [Typecasting data](/en-us/netscaler/11-1/appexpert/policies-and-expressions/ns-typecasting-data-wrapper-con.html).
HTTP.REQ.FULL_HEADER	Returns the contents of the complete set of HTTP header fields including the request line (for example, “GET /brochures/index.html HTTP/1.1”) and the terminating \r\n\r\n sequence.

HTTP Header Prefix	Description
HTTP.REQ.DATE	Returns the contents of the HTTP Date header. The following date formats are recognized: RFC822. Sun, 06 Jan 1980 8:49:37 GMT, RFC850. Sunday, 06-Jan-80 9:49:37 GMT, ASCII TIME. Sun Jan 6 08:49:37 1980. To evaluate a Date header as a date object, see Default Syntax Expressions: Working with Dates, Times, and Numbers.
HTTP.REQ.COOKIE	(Name/Value List) Returns the contents of the HTTP Cookie header.
HTTP.REQ.TXID	Returns the HTTP transaction ID. The value is a function of an internal transaction number, system boot time and system MAC address.
HTTP.RES.HEADER(“”)	Returns the contents of the HTTP header specified by the argument. The header name cannot exceed 32 characters.

HTTP Header Prefix	Description
HTTP.RES.FULL_HEADE	Returns the contents of the complete set of HTTP header fields including the status line (for example, “HTTP/1.1 200 OK”) and the terminating \r\n\r\n sequence.
HTTP.RES.SET_COOKIE or HTTP.RES.SET_COOKIE2	Returns the Set-Cookie header object in a response.
HTTP.RES.SET_COOKIE	Returns the cookie of the specified name if it is present. If it is not present, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the specified cookie was not found in these headers.

HTTP Header Prefix	Description	
HTTP.RES.SET_COOKIE(<name>).DOMAIN	Returns the value of the first Domain field in the cookie. For example, if the cookie is Set-Cookie : Customer = “ABC”; DOMAIN=”.abc.com”; DOMAIN=.xyz.com, the following expression returns .abc.com: http.res.set_cookie.cookie(“customer”).domain. A string of zero length is returned if the Domain field or its value is absent.	HTTP.RES.SET_COOKIE2(<name>).DOMAIN
HTTP.RES.SET_COOKIE	Returns a Boolean TRUE if a Cookie with the name specified in the <name> argument exists in the Set-Cookie header. This prefix returns UNDEF if more than 15 Set-Cookie headers are present and the named cookie is not in the first 15 headers.	

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE(<name>).EXPIRES or HTTP.RES.SET_COOKIE2.COOKIE(<name>).EXPIRES	Returns the Expires field of the cookie. This is a date string that can be evaluated as a number, as a time object, or as text. If multiple Expires fields are present, the first one is returned. If the Expires field is absent, a text object of length zero is returned. To evaluate the returned value as a time object, see Default Syntax Expressions: Working with Dates, Times, and Numbers.
HTTP.RES.SET_COOKIE	PATH.GET(n)

HTTP Header Prefix	Description	
o HTTP.RES.SET_COOKIE2.COOKIE(<name>).PATH	PATH.GET(n)	Returns the value of Path field of the cookie as a slash- ("/") separated list. Multiple instances of a slash are treated as single slash. If multiple Path fields are present, the value of the first instance is returned. For example, the following is a cookie with two path fields:Set-Cookie : Customer = "ABC"; PATH="/a//b/c"; PATH= "/x/y/z". The following expression returns /a//b/c from this cookie: http.res.set_cookie.cookie("Customer").path. The following expression returns b: http.res.set_cookie.cookie("Customer").path.get(2). Quotes are stripped from the returned value. A string of zero length is returned if the Path field or its value is absent.

HTTP Header Prefix	Description	
HTTP.RES.SET_COOKIE	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header("Cus following expression returns a value of 5: http.req.header("Cus	KIE2.COOKIE ents.count. T

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE(<name>).PORT or HTTP.RES.SET_COOKIE2.COOKIE(<name>).PORT	Returns the value of Port field of the cookie. Operate as a comma-separated list. For example, the following expression returns 80. 2580 from Set-Cookie : Customer = “ABC”; PATH=”/a/b/c”; PORT= “80, 2580”: http.res.set_cookie.cookie(“ABC”).port. A string of zero length is returned if the Port field or value is absent.

HTTP Header Prefix	Description	
HTTP.RES.SET_COOKIE	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element.As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header(“Cus following expression returns a value of 5: http.req.header(“Cus	KIE2.COOKIE nents.count. T

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE(<name>).VERSION or HTTP.RES.SET_COOKIE2.COOKIE(<name>).VERSION	Returns the value of the first Version field in the cookie as a decimal integer. For example, the following expression returns 1 from the cookie Set-Cookie : Customer = “ABC”; VERSION = “1”; VERSION = “0”: http.res.set_cookie.cookie(“CUSTOMER”).version. A zero is returned if the Version field or its value is absent or if the value is not a decimal number.
HTTP.RES.SET_COOKIE <integer>) or HTTP.RES. <integer>)	Returns the nth instance (0-based) of the cookie with the specified name. If the cookie is absent, returns a text object of length 0. Returns UNDEF if more than 15 Set-Cookie headers are present and the cookie is not found.

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).DOMAIN or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).DOMAIN	Returns the value of the Domain field of the first cookie with the specified name. For example, the following expression returns a value of abc.com from the cookie Set-Cookie : Customer = “ABC”; DOMAIN=”.abc.com”; DOMAIN=.xyz.com: http.res.set_cookie.cookie(“CUSTOMER”).domain. A string of zero length is returned if the Domain field or its value is absent.
HTTP.RES.SET_COOKIE <integer>).EXPIRES or H <integer>).EXPIRES	Returns the nth instance (0-based) of the Expires field of the cookie with the specified name as a date string. The value can be operated upon as a time object that supports a number of date formats. If the Expires attribute is absent a string of length zero is returned.

HTTP Header Prefix	Description	
HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PATH	PATH.GET(i) or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).PATH PATH.GET(i)	Returns the value of the Path field of the nth cookie, as a ‘/’ separated list. Multiple /s are treated as a single /. For example, the following expression returns /a//b/c from the cookie Set-Cookie : Customer = “ABC”; PATH=”/a//b/c”; PATH= ”/x/y/z”: http.res.set_cookie.c following returns b: http.res.set_cookie.cookie(“CUST string of zero length is returned if the Path field or its value is absent.

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE <integer>).PATH.IGNOR <integer>).PATH.IGNOR	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123„24, ,15 the following expression returns a value of 4: http.req.header(“Cus ents.count. T following expression returns a value of 5: http.req.header(“Cus

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).PORT or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).PORT	Returns the value or values of the Port field of the named cookie as a ‘,’ separated list. For example, the following expression returns 80, 2580 from the cookie Set-Cookie : Customer = “ABC”; PATH=”/a/b/c”; PORT= “80, 2580”: http.res.set_cookie.cookie(“ABC”).port. A string of zero length is returned if the Port field or its value is absent.

HTTP Header Prefix	Description
HTTP.RES.SET_COOKIE <integer>).PORT.IGNOR <integer>).PORT.IGNOR	Ignores the empty elements in the list. For example, in the list a=10,b=11, ,c=89, the element delimiter in the list is , and the list has an empty element following a=10. The element following b=11 is not considered an empty element. As another example, in the following expression, if a request contains Cust_Header : 123,,24, ,15 the following expression returns a value of 4: http.req.header(“Cus ents.count. T following expression returns a value of 5: http.req.header(“Cus
HTTP.RES.SET_COOKIE.COOKIE(<name>, <integer>).VERSION or HTTP.RES.SET_COOKIE2.COOKIE(<name>, <integer>).VERSION	Returns the value of Version field of the nth cookie as a decimal integer. A string of zero length is returned if the Port field or its value is absent.

HTTP Header Prefix	Description
HTTP.RES.TXID	Returns the HTTP transaction ID. The value is a function of an internal transaction number, system boot time and system MAC address.

Operations for HTTP headers

The following table describes operations that you can specify with the prefixes for HTTP headers.

HTTP Header Operation	Description
http header .EXISTS	Returns a Boolean TRUE if an instance of the specified header type exists. Following is an example: http.req.header("Cache-Control").exists
http header.CONTAINS" http header.CONTAINS(<string>)	Returns a Boolean TRUE if the <string> argument appears in any instance of the header value. Note: This operation overrides any text-based Contains operations on all instances of the current header type.
http header .COUNT	Returns the number of headers in a request or response, to a maximum of 15 headers of the same type. The result is undefined if there are more than 15 instances of the header.
http header.AFTER_STR(<string>)	Extracts the text that follows the first occurrence of the <string> argument. The headers are evaluated from the last instance to the first.
http header.BEFORE_STR(<string>)	Extracts the text that appears prior to the first occurrence of the input <string> argument. The headers are evaluated from the last instance to the first.

HTTP Header Operation	Description
http header.INSTANCE(<instance number>)	An HTTP header can occur multiple times in a request or a response. This operation returns the header that occurs <instance number> of places before the final instance. For example, instance(0) selects the last instance of the current type, instance(1) selects the next-to-last instance, and so on. This prefix cannot be used in bidirectional policies.
http header.SUBSTR(<string>)	Extracts the text that matches the <string> argument. The headers are evaluated from the last instance to the first.
http header.VALUE(<instance number>)	An HTTP header can occur multiple times in a request or a response. VALUE(0) selects the value in the last instance, VALUE(1) selects the value in the next-to-last instance, and so on. The <instance number> argument cannot exceed 14.

Prefixes for cache-control headers

The following prefixes apply specifically to Cache-Control headers.

HTTP Header Prefix	Description
HTTP.REQ.CACHE_CONTROL	Returns a Cache-Control header in an HTTP request.
HTTP.RES.CACHE_CONTROL	Returns a Cache-Control header in an HTTP response.

Operations for cache-control headers

You can apply any of the operations for HTTP headers to Cache-Control headers. For more information, see Operations for HTTP headers.

In addition, the following operations identify specific types of Cache-Control headers. See RFC 2616 for information about these header types.

HTTP Header Operation	Description
Cache-Control header.NAME(<integer>)	Returns as a text value the name of the Cache-Control header that corresponds to the nth component in a name-value list, as specified by <integer>. The index of the name-value component is 0-based. If the <integer> that is specified by the integer argument is greater than the number of components in the list, a zero-length text object is returned. Following is an example: http.req.cache_control.name(3).contains("some_text")
Cache-Control header.IS_INVALID	Returns a Boolean TRUE if the Cache-Control header is not present in the request or response. Following is an example: http.req.cache_control.is_invalid
Cache-Control header.IS_PRIVATE	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: http.req.cache_control.is_private
Cache-Control header.IS_PUBLIC	Returns a Boolean TRUE if the Cache-Control header has the value Private. Following is an example: http.req.cache_control.is_public
Cache-Control header.IS_NO_STORE	Returns a Boolean TRUE if the Cache-Control header has the value No-Store. Following is an example: http.req.cache_control.is_no_store
Cache-Control header.IS_NO_CACHE	Returns a Boolean TRUE if the Cache-Control header has the value No-Cache. Following is an example: http.req.cache_control.is_no_cache
Cache-Control header.IS_MAX_AGE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Age. Following is an example: http.req.cache_control.is_max_age
Cache-Control header.IS_MIN_FRESH	Returns a Boolean TRUE if the Cache-Control header has the value Min-Fresh. Following is an example: http.req.cache_control.is_min_fresh
Cache-Control header.IS_MAX_STALE	Returns a Boolean TRUE if the Cache-Control header has the value Max-Stale. Following is an example: http.req.cache_control.is_max_stale

HTTP Header Operation	Description
Cache-Control header.IS_MUST_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Must-Revalidate. Following is an example: http.req.cache_control.is_must_revalidate
Cache-Control header.IS_NO_TRANSFORM	Returns a Boolean TRUE if the Cache-Control header has the value No-Transform. Following is an example: http.req.cache_control.is_no_transform
Cache-Control header.IS_ONLY_IF_CACHED	Returns a Boolean TRUE if the Cache-Control header has the value Only-If-Cached. Following is an example: http.req.cache_control.is_only_if_cached
Cache-Control header.IS_PROXY_REVALIDATE	Returns a Boolean TRUE if the Cache-Control header has the value Proxy-Revalidate. Following is an example: http.req.cache_control.is_proxy_revalidate
Cache-Control header.IS_S_MAXAGE	Returns a Boolean TRUE if the Cache-Control header has the value S-Maxage. Following is an example: http.req.cache_control.is_s_maxage
Cache-Control header.IS_UNKNOWN	Returns a Boolean TRUE if the Cache-Control header is of an unknown type. Following is an example: http.req.cache_control.is_unknown
Cache-Control header.MAX_AGE	Returns the value of the Cache-Control header Max-Age. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.max_age.le(3)
Cache-Control header.MAX_STALE	Returns the value of the Cache-Control header Max-Stale. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.max_stale.le(3)
Cache-Control header.MIN_FRESH	Returns the value of the Cache-Control header Min-Fresh. If this header is absent or invalid, 0 is returned. Following is an example: http.req.cache_control.min_fresh.le(3)

HTTP Header Operation	Description
Cache-Control header.S_MAXAGE	Returns the value of the Cache-Control header S-Maxage. If this header is absent or invalid, 0 is returned.Following is an example: http.req.cache_control.s_maxage.eq(2)

Expressions for extracting segments of URLs

September 21, 2020

You can extract URLs and portions of URLs, such as the host name, or a segment of the URL path. For example, the following expression identifies HTTP requests for image files by extracting image file suffixes from the URL:

```
http.req.url.suffix.eq("jpeg")|| http.req.url.suffix.eq("gif")
```

Most expressions for URLs operate on text and are described in Expression Prefixes for Text in HTTP Requests and Responses. This section discusses the GET operation. The GET operation extracts text when used with the following prefixes:

- HTTP.REQ.URL.PATH
- VPN.BASEURL.PATH
- VPN.CLIENTLESS_BASEURL.PATH

The following table describes prefixes for HTTP URLs.

URL Prefix	Description
HTTP.REQ.URL.PATH.GET(<n>)	Returns a slash- ("/") separated list from the URL path. For example, consider the following URL: `http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1`. The following expression returns dir1 from this URL: <http.req.url.path.get(1)>. The following expression returns dir2: http.req.url.path.get(2)

URL Prefix	Description
HTTP.REQ.URL.PATH.GET_REVERSE(<n>)	Returns a slash- (“/”) separated list from the URL path, starting from the end of the path. For example, consider the following URL: `http://www.mycompany.com/dir1/dir2/dir3/index.html?a=1`. The following expression returns index.html from this URL: <http.req.url.path.get_reverse(0)>. The following expression returns dir3: http.req.url.path.get_reverse(1)

Expressions for HTTP status codes and numeric HTTP payload data other than dates

September 21, 2020

The following table describes prefixes for numeric values in HTTP data other than dates.

Prefix	Description
HTTP.REQ.CONTENT_LENGTH	Returns the length of an HTTP request as a number. Following is an example: http.req.content_length < 500
HTTP.RES.CONTENT_LENGTH	Returns the length of the HTTP response as a number. Following is an example: http.res.content_length <= 1000
HTTP.RES.STATUS	Returns the response status code
HTTP.RES.IS_REDIRECT	Returns a Boolean TRUE if the response code is associated with a redirect. Following are the redirect response codes: 300 (Multiple Choices), 301 (Moved Permanently), 302 (Found), 303 (See Other), 305 (Use Proxy), and 307 (Temporary Redirect). Note: Status code 304 is not considered a redirect HTTP response status code. Status code 306 is unused.

SIP expressions

September 21, 2020

The NetScaler Advanced policy expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions are intended to be used in policies for any supported protocol that operates on a request/response basis. These expressions can be used in content switching, rate limiting, responder, and rewrite policies.

Certain limitations apply to SIP expressions used with responder policies. Only the DROP, NOOP or RESPONDWITH actions are allowed on a SIP load balancing virtual server. Responder policies can be bound to a load balancing virtual server, an override global bind point, a default global bind point, or a sip_udp policy label.

The header format used by the SIP protocol is similar to that used by the HTTP protocol, so many of the new expressions look and function much like their HTTP analogs. Each SIP header consists of a line that includes the SIP method, the URL, and the version, followed by a series of name-value pairs that look like HTTP headers.

Following is a sample SIP header that is referred to in the expressions tables beneath it:

```
INVITE sip:16@www.sip.com:5060;transport=udp SIP/2.0
Record-Route: <sip:200.200.100.22;lr=on>
Via: SIP/2.0/UDP 200.200.100.22;branch=z9hG4bK444b.c8e103d1.0;rport=5060;
    received=10.102.84.18
Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;
    received=10.102.84.160
From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
To: "16" <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185
Call-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180
Max-Forwards: 69CSeq: 101 INVITE
User-Agent: Cisco-CP7940G/8.0
Contact: <sip:12@10.102.84.180:5060;transport=udp>
Expires: 180
Accept: application/sdp
Allow: ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,UPDATE
Supported: replaces,join,norefersub
Content-Length: 277
Content-Type: application/sdp
Content-Disposition: session;handling=optiona
<!--NeedCopy-->
```

SIP reference tables

The following tables contain lists of expressions that operate on SIP headers. The first table contains expressions that apply to request headers. Most response-based expressions are nearly the same as the corresponding request-based expressions. To create a response expression from the corresponding request expression, you change the first two sections of the expression from SIP.REQ to SIP.RES, and make other obvious adjustments. The second table contains those response expressions that are unique to responses and have no request equivalents. You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

SIP request expressions

Expression	Description
SIP.REQ.METHOD	Operates on the method of the SIP request. The supported SIP request methods are ACK, BYE, CANCEL, INFO, INVITE, MESSAGE, NOTIFY, OPTIONS, PRACK, PUBLISH, REFER, REGISTER, SUBSCRIBE, and UPDATE. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns INVITE.
SIP.REQ.URL	Operates on the SIP request URL. This expression is a derivative of the text class, so all operations that are applicable to text are applicable to this method. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returnssip:16@10.102.84.181:5060;transport=udp.
SIP.REQ.URL.PROTOCOL	Returns the URL protocol. For example, for a SIP URL ofsip:16@www.sip.com:5060;transport=udp, this expression returns sip.

Expression	Description
SIP.REQ.URL.HOSTNAME	Returns the hostname portion of the SIP URL. For example, for a SIP URL ofsip:16@www.sip.com:5060;transport=udp, this expression returns www.sip.com:5060.
SIP.REQ.URL.HOSTNAME.PORT	Returns the port portion of the SIP URL hostname. If no port is specified, this expression returns the default SIP port, 5060. For example, for a SIP hostname of www.sip.com:5060, this expression returns 5060.
SIP.REQ.URL.HOSTNAME.DOMAIN	Returns the domain name portion of the SIP URL hostname. If the host is an IP address, then this expression returns an incorrect result. For example, for a SIP hostname of www.sip.com:5060, this expression returns sip.com. For a SIP hostname of 192.168.43.15:5060, this expression returns an error.
SIP.REQ.URL.HOSTNAME.SERVER	Returns the server portion of the host. For example, for a SIP hostname of www.sip.com:5060, this expression returns www.
SIP.REQ.URL.USERNAME	Returns the username that precedes the @ character. For example, for a SIP URL of sip:16@www.sip.com:5060;transport=udp, this expression returns 16.
SIP.REQ.VERSION	Returns the SIP version number in the request. For example, for a SIP request of INVITE sip:16@10.102.84.181:5060;transport=udp SIP/2.0, this expression returns SIP/2.0.
SIP.REQ.VERSION.MAJOR	Returns the major version number (the number to the left of the period). For example, for a SIP version number of SIP/2.0, this expression returns 2.

Expression	Description
SIP.REQ.VERSION.MINOR	Returns the minor version number (the number to the right of the period). For example, for a SIP version number of SIP/2.0, this expression returns 0.
SIP.REQ.CONTENT_LENGTH	Returns the contents of the Content-Length header. This expression is a derivative of thesip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Content-Length header of Content-Length: 277, this expression returns 277.
SIP.REQ.TO	Returns the contents of the To header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns ”16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.TO.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:16@sip_example.com.
SIP.REQ.TO.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 16.
SIP.REQ.TO.TAG	Returns the “tag” vale from the “tag” name value pair in the TO header. For example, for a SIP To header of To: “16” <sip:16@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.

Expression	Description
SIP.REQ.FROM	Returns the contents of the From header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.ADDRESS	Returns the SIP URI, which is found in the sip_url object. All operations that are available for SIP URIs can be used. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns sip:12@sip_example.com.
SIP.REQ.FROM.DISPLAY_NAME	Returns the display name portion of the To header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 12.
SIP.REQ.FROM.TAG	Returns the "tag" value from the "tag" name/value pair in the TO header. For example, for a SIP From header of From: "12" <sip:12@sip_example.com>;tag=00127f54ec85a6d90cc14f45-53cc0185, this expression returns 00127f54ec85a6d90cc14f45-53cc0185.
SIP.REQ.VIA	Returns the complete Via header. If there are multiple Via headers in the request, returns the last Via header. For example, for the two Via headers in the sample SIP header, this expression returns Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re
SIP.REQ.VIA.SENTBY_ADDRESS	Returns the address that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 10.102.84.180.

Expression	Description
SIP.REQ.VIA.SENTBY_PORT	Returns the port that sent the request. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 5060.
SIP.REQ.VIA.RPORT	Returns the value from the rport name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 5060.
SIP.REQ.VIA.BRANCH	Returns the value from the branch name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns z9hG4bK03e76d0b.
SIP.REQ.VIA.RECEIVED	Returns the value from the received name/value pair. For example, for the Via header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re this expression returns 10.102.84.160.
SIP.REQ.CALLID	Returns the contents of the Callid header. This expression is a derivative of the sip_header_t class, so all operations that are available for SIP headers can be used. For example, for a SIP Callid header ofCall-ID: 00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180, this expression returns00127f54-ec850017-0e46f5b9-5ec149c2@10.102.84.180.
SIP.REQ.CSEQ	Returns the CSEQ number from the CSEQ, as an integer. For example, for a SIP CSEQ header of CSeq: 101 INVITE, this expression returns 101.

Expression	Description
SIP.REQ.HEADER(<header_name>)	Returns the specified SIP header. For , substitute the name of the header that you want. For example, to return the SIP From header, you would type SIP.REQ.HEADER(”From”).
SIP.REQ.HEADER(<header_name>).INSTANCE()	Returns the specified instance of the specified SIP header. Multiple instances of the same SIP header can occur. Where you want a specific instance of such a SIP header (for example, a specific Via header), you can specify that header by typing a number as the . Header instances are matched from last (0) to first. In other words, SIP.REQ.HEADER(”Via”).INSTANCE(0) returns the last instance of the Via header, while SIP.REQ.HEADER(”Via”).INSTANCE(1) returns the last instance but one of the Via header, and so on. For example, if used on the example SIP header, SIP.REQ.HEADER(”Via”).INSTANCE(1) returnsVia: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.
SIP.REQ.HEADER(<header_name>).VALUE()	Returns the contents of the specified instance of the specified SIP header. The usage is nearly the same as the previous expression. For example, if used on the SIP header example in the preceding table entry, SIP.REQ.HEADER(“Via”).VALUE(1) returns SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060.
SIP.REQ.HEADER(<header_name>).COUNT	Returns the number of instances of a particular header as an integer. For example, if used on the SIP header example above, SIP.REQ.HEADER(“Via”).COUNT returns 2.

Expression	Description
SIP.REQ.HEADER(<header_name>).EXISTS	Returns a boolean value of true or false, depending upon whether the specified header exists or not. For example, if used on the SIP header example above, SIP.REQ.HEADER("Expires").EXISTSreturns true, while SIP.REQ.HEADER("Caller-ID").EXISTS returns false.
SIP.REQ.HEADER(<header_name>).LIST	Returns the comma-separated parameter list in the specified header. For example, if used on the SIP header example above, SIP.REQ.HEADER("Allow").LIST returns ACK,BYE,CANCEL,INVITE,NOTIFY,OPTIONS,REFER,REGISTER,U You can append the string .GET() to select a specific list item. For example, to get the first item (ACK) from the above list, you would type SIP.REQ.HEADER("Allow").LIST.GET(0). To extract the second item (BYE), you would type SIP.REQ.HEADER("Allow").LIST.GET(1). Note: If the specified header contains a list of name/value pairs, the entire name/value pair is returned.
SIP.REQ.HEADER(<header_name>).TYPECAST_SI	Typecasts to . Any text can be typecasted to thesip_header_t class, after which all header-based operations can be used. After you perform this operation, you can apply all operations that can be used with . For example, the expression SIP.REQ.CONTENT_LENGTH.TYPECAST_SIP_HEADER_T ty casts all instances of the Content-Length header. After you perform this operation, you can apply all header operations to all instances of the specified header.
SIP.REQ.HEADER(<header_name>).CONTAINS(<string>)	Returns boolean true if the specified text string is present in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.

Expression	Description
SIP.REQ.HEADER(<header_name>).EQUALS_ANY	Returns boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).CONTAINS_ANY(<patset>)	Returns Boolean true if any pattern associated with <patset> matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).CONTAINS_IN	Returns the index of the matching pattern associated with <patset> if that pattern matches any content in any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).EQUALS_INDEX(<patset>)	Returns the index of the matching pattern associated with <patset> if that pattern matches any instance of the specified header. Operates on all the instances of the specified header. Header instances are matched from last (0) to first.
SIP.REQ.HEADER(<header_name>).SUBSTR(<stri	If the specified string is present in any instance of the specified header, this expression returns that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re turns “rport=5060”.SIP.REQ.HEADER(“Via”).SUBSTR(“rport=50 turns an empty string.

Expression	Description
SIP.REQ.HEADER(<header_name>).AFTER_STR(<string>)	If the specified string is present in any instance of the specified header, this expression returns the string immediately after that string. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER("Via").AFTER_STR("rport=") returns 5060.
SIP.REQ.HEADER(<header_name>).REGEX_MATC	Returns boolean true if the specified regular expression (regex) matches any instance of the specified header. You must specify the regular expression in the following format: re<delimiter>regular expression<same delimiter>. The regular expression cannot be larger than 1499 characters in length. It must conform to the PCRE regular expression library. See http://www.pcre.org/pcre.txt for documentation on PCRE regular expression syntax. The pcrepattern man page also has useful information on specifying patterns by using PCRE regular expressions. The regular expression syntax supported in this expression has some differences from PCRE. Back references are not allowed. You should avoid recursive regular expressions; although some work, many do not. The dot (.) metacharacter matches newlines. Unicode is not supported.SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression.

Expression	Description
SIP.REQ.HEADER(<header_name>).REGEX_SELECT(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).REGEX_SELECT(“received=[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) returns received=10.102.84.160.
SIP.REQ.HEADER(<header_name>).AFTER_REGE	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately after that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).AFTER_REGEX(“received=”) returns10.102.84.160.
SIP.REQ.HEADER(<header_name>).BEFORE_REGEX(<regex>)	If the specified regex matches any text in any instance of the specified header, this expression returns the string immediately before that text. For example, for the SIP header Via: SIP/2.0/UDP 10.102.84.180:5060;branch=z9hG4bK03e76d0b;rport=5060;re the expression SIP.REQ.HEADER(“Via”).BEFORE_REGEX(“[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}.[0-9]{1,3}”) returns received=.
SIP.REQ.FULL_HEADER	Returns the entire SIP header, including the terminating CR/LF.
SIP.REQ.IS_VALID	Returns boolean true if the request format is valid.
SIP.REQ.BODY(<length>)	Returns the request body, up to the specified length. If the specified length is greater than the length of the request body, this expression returns the entire request body.

Expression	Description
SIP.REQ.LB_VSERVER	Returns the name of the load balancing virtual server (LB vserver) that is serving the current request.
SIP.REQ.CS_VSERVER	Returns the name of the content switching virtual server (CS vserver) that is serving the current request.

SIP response expressions

Expression	Description
SIP.RES.STATUS	Returns the SIP response status code. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns 100.
SIP.RES.STATUS_MSG	Returns the SIP response status message. For example, if the first line of the response is SIP/2.0 100 Trying, this expression returns Trying.
SIP.RES.IS_REDIRECT	Returns boolean true if the response code is a redirect.
SIP.RES.METHOD	Returns the response method extracted from the request method string in the CSeq header.

Operations for HTTP, HTML, and XML encoding and "safe" characters

September 21, 2020

The following operations work with the encoding of HTML data in a request or response and XML data in a POST body.

- **<text>.HTML_XML_SAFE:**
  Transforms special characters into XML safe format, as in the following examples:

  A left-pointing angle bracket (<) is converted to <
  A right-pointing angle bracket (>) is converted to >
  An ampersand (&) is converted to &

This operation safeguards against cross-site scripting attacks. Maximum length of the transformed text is 2048 bytes. This is a read-only operation.

After applying the transformation, additional operators that you specify in the expression are applied to the selected text. Following is an example:

http.req.url.query.html_xml_safe. contains("myQueryString")

- **<text>.HTTP_HEADER_SAFE:**
  Converts all new line ('\n') characters in the input text to '%0A' to enable the input to be used safely in HTTP headers.

  This operation safeguards against response-splitting attacks.

  The maximum length of the transformed text is 2048 bytes. This is a read-only operation.

- **<text>.HTTP_URL_SAFE:**
  Converts unsafe URL characters to '%xx' values, where "xx" is a hex-based representation of the input character. For example, the ampersand (&) is represented as %26 in URL-safe encoding. The maximum length of the transformed text is 2048 bytes. This is a read-only operation.

  Following are URL safe characters. All others are unsafe:

  - Alpha-numeric characters: a-z, A-Z, 0-9
  - Asterix: "*"
  - Ampersand: "&"
  - At-sign: "@"
  - Colon: ":"
  - Comma: ","
  - Dollar: "$"
  - Dot: "."
  - Equals: "="
  - Exclamation mark: "!"
  - Hyphen: "-"
  - Open and close parentheses: "(", ")"
  - Percent: "%"
  - Plus: "+"
  - Semicolon: ";"
  - Single quote: "'"
  - Slash: "/"
  - Question mark: "?"
  - Tilde: "~"
  - Underscore: "_"

- **<text>.MARK_SAFE:**
  Marks the text as safe without applying any type of data transformation.

- **<text>.SET_TEXT_MODE(URLENCODED|NOURLENCODED)
  Transforms all %HH encoding in the byte stream. This operation works with characters (not bytes). By default, a single byte represents a character in ASCII encoding. However, if you specify URLENCODED mode, three bytes can represent a character.

  In the following example, a PREFIX(3) operation selects the first 3 characters in a target.

  `http.req.url.hostname.prefix(3)`

  In the following example, the NetScaler can select up to 9 bytes from the target:

  `http.req.url.hostname.set_text_mode(urlencoded).prefix(3)`

- **<text>.SET_TEXT_MODE(PLUS_AS_SPACE|NO_PLUS_AS_SPACE):**
  Specifies how to treat the plus character (+). The PLUS_AS_SPACE option replaces a plus character with white space. For example, the text "hello+world" becomes "hello world." The NO_PLUS_AS_SPACE option leaves plus characters as they are.

- **<text>.SET_TEXT_MODE(BACKSLASH_ENCODED|NO_BACKSLASH_ENCODED):**
  Specifies whether or not backslash decoding is performed on the text object represented by <text>.

  If BACKSLASH_ENCODED is specified, the SET_TEXT_MODE operator performs the following operations on the text object:

  - All occurrences of "\XXX" will be replaced with the character "Y" (where XXX represents a number in the octal system and Y represents the ASCII equivalent of XXX). The valid range of octal values for this type of encoding is 0 to 377. For example, the encoded text "http\72//" and http\072//" will both be decoded to http://, where the colon (:) is the ASCII equivalent of the octal value "72".
  - All occurrences of "\xHH" will be replaced with the character "Y" (HH represents a number in the hexadecimal system and Y denotes the ASCII equivalent of HH. For example, the encoded text "http\x3a//" will be decoded to http://, where the colon (:) is the ASCII equivalent of the hexadecimal value "3a".
  - All occurrences of "\uWWXX" will be replaced with the character sequence "YZ" (Where WW and XX represent two distinct hexadecimal values and Y and Z represent their ASCII equivalents of WW and XX respectively. For example, the encoded text "http%u3a2f/" and "http%u003a//" will both be decoded to http://, where "3a" and "2f" are two hexadecimal values and the colon (:) and forward slash ("/") represent their ASCII equivalents respectively.
  - All occurrences of "\b", "\n", "\t", "\f", and "\r" are replaced with the corresponding ASCII characters.
    If NO_BACKSLASH_ENCODED is specified, backslash decoding is not performed on the text object.

- **<text>.SET_TEXT_MODE(BAD_ENCODE_RAISE_UNDEF|NO_BAD_ENCODE_RAISE_UNDEF):** Performs the associated undefined action if either the URLENCODED or the BACKSLASH_ENCODED mode is set and bad encoding corresponding to the specified encoding mode is encountered in the text object represented by <text>.

  If NO_BAD_ENCODE_RAISE_UNDEF is specified, the associated undefined action will not be performed when bad encoding is encountered in the text object represented by<text>.

Expressions for TCP, UDP, and VLAN data

September 21, 2020

TCP and UDP data take the form of a string or a number. For expression prefixes that return string values for TCP and UDP data, you can apply any text-based operations. For more information, see Advanced policy expressions: Evaluating text.

For expression prefixes that return numeric value, such as a source port, you can apply an arithmetic operation. For more information, see Basic operations on expression prefixes and Compound operations for numbers.

The following table describes prefixes that extract TCP and UDP data.

GET Operation	Description
CLIENT.TCP.PAYLOAD(<integer>)	Returns TCP payload data as a string, starting with the first character in the payload and continuing for the number of characters in the <integer> argument. You can apply any text-based operation to this prefix.
CLIENT.TCP.SRCPORT	Returns the ID of the current packet's source port as a number.
CLIENT.TCP.DSTPORT	Returns the ID of the current packet's destination port as a number.
CLIENT.TCP.OPTIONS	Returns the TCP options set by the client. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the server, see the SERVER.TCP.OPTIONS prefix.

GET Operation	Description
CLIENT.TCP.OPTIONS.COUNT	Returns the number of TCP options that the client has set.
CLIENT.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or option kind) is specified as the argument. The value is returned as a string of bytes in big endian format (or network byte order). Parameters: type - Type value
CLIENT.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). Parameters: m - TCP option enumeration constant
CLIENT.TCP.REPEATER_OPTION.EXISTS	Returns a Boolean TRUE if Repeater TCP options exist.
CLIENT.TCP.REPEATER_OPTION.IP	Returns the branch repeater’s IPv4 address from the Repeater TCP options.
CLIENT.TCP.REPEATER_OPTION.MAC	Returns the branch repeater’s MAC address from the Repeater TCP options.
CLIENT.UDP.DNS.DOMAIN	Returns the DNS domain name.
CLIENT.UDP.DNS.DOMAIN.EQ(“<hostname>”)	Returns a Boolean TRUE if the domain name matches the <hostname> argument. The comparison is case insensitive. Following is an example: client.udp.dns.domain.eq(“www.mycompany.com”)
CLIENT.UDP.DNS.IS_AAAAREC	Returns a Boolean TRUE if the record type is AAAA. These types of records indicate an IPv6 address in forward lookups.
CLIENT.UDP.DNS.IS_ANYREC	Returns a Boolean TRUE if it is of any record type.

GET Operation	Description
CLIENT.UDP.DNS.IS_AREC	Returns a Boolean TRUE if the record is type A. Type A records provide the host address.
CLIENT.UDP.DNS.IS_CNAMEREC	Returns a Boolean TRUE if the record is of type CNAME. In systems that use multiple names to identify a resource, there is one canonical name and a number of aliases. The CNAME provides the canonical name.
CLIENT.UDP.DNS.IS_MXREC	Returns a Boolean TRUE if the record is of type MX (mail exchanger). This DNS record describes a priority and a host name. The MX records for the same domain name specify the email servers in the domain and the priority for each server.
CLIENT.UDP.DNS.IS_NSREC	Returns a Boolean TRUE if the record is of type NS. This is a name server record that includes a host name with an associated A record. This enables locating the domain name that is associated with the NS record.
CLIENT.UDP.DNS.IS_PTRREC	Returns a Boolean TRUE if the record is of type PTR. This is a domain name pointer and is often used to associate a domain name with an IPv4 address.
CLIENT.UDP.DNS.IS_SOAREC	Returns a Boolean TRUE if the record is of type SOA. This is a start of authority record.
CLIENT.UDP.DNS.IS_SRVREC	Returns a Boolean TRUE if the record is of type SRV. This is a more general version of the MX record.
CLIENT.UDP.DSTPORT	Returns the numeric ID of the current packet's UDP destination port.
CLIENT.UDP.SRCPORT	Returns the numeric ID of the current packet's UDP source port.
CLIENT.UDP.RADIUS	Returns RADIUS data for the current packet.
CLIENT.UDP.RADIUS.ATTR_TYPE(<type>)	Returns the value for the attribute type specified as the argument.
CLIENT.UDP.RADIUS.USERNAME	Returns the RADIUS user name.

GET Operation	Description
CLIENT.TCP.MSS	Returns the maximum segment size (MSS) for the current connection as a number.
CLIENT.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.
SERVER.TCP.DSTPORT	Returns the numeric ID of the current packet's destination port.
SERVER.TCP.SRCPORT	Returns the numeric ID of the current packet's source port.
SERVER.TCP.OPTIONS	Returns the TCP options set by the server. Examples of TCP options are Maximum Segment Size (MSS), Window Scale, Selective Acknowledgements (SACK), and Time Stamp Option. The COUNT, TYPE(<type>), and TYPE_NAME(<m>) operators can be used with this prefix. For the TCP options set by the client, see the CLIENT.TCP.OPTIONS prefix.
SERVER.TCP.OPTIONS.COUNT	Returns the number of TCP options that the server has set.
SERVER.TCP.OPTIONS.TYPE(<type>)	Returns the value of the TCP option whose type (or option kind) is specified as the argument. The value is returned as a string of bytes in big endian format (or network byte order). Parameters: type - Type value
SERVER.TCP.OPTIONS.TYPE_NAME(<m>)	Returns the value of the TCP option whose enumeration constant is specified as the argument. The enumeration constants that you can pass as the argument are REPEATER, TIMESTAMP, SACK_PERMITTED, WINDOW, and MAXSEG. To specify the TCP option kind instead of these enumeration constants, use CLIENT.TCP.OPTIONS.TYPE(<type>). For other TCP options, you must use CLIENT.TCP.OPTIONS.TYPE(<type>). Parameters: m - TCP option enumeration constant

GET Operation	Description
SERVER.VLAN	Operates on the VLAN through which the current packet entered the NetScaler.
SERVER.VLAN.ID	Returns the numeric ID of the VLAN through which the current packet entered the NetScaler.

Expressions for evaluating a DNS message and identifying its carrier protocol

September 21, 2020

You can evaluate DNS requests and responses by using expressions that begin with DNS.REQ and DNS.RES, respectively. You can also identify the transport layer protocol that is being used to send the DNS messages.

The following functions return the contents of a DNS query.

Function	Description
DNS.REQ.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS query. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.
DNS.REQ.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the DNS query. The field indicates the type of resource record (for example, A, NS, or CNAME) for which the name server is being queried. The returned value can be compared to one of the following values by using the EQ() and NE()functions: A, AAAA, NS, SRV, PTR, CNAME, SOA, MX, and ANY. Note: You can use only the EQ() and NE() functions with the TYPE function. Example: DNS.REQ.QUESTION.TYPE.EQ(MX)

The following functions return the contents of a DNS response.

Function	Description
DNS.RES.HEADER.RCODE	Return the response code (the value of the RCODE field) in the header section of the DNS response. You can use only the EQ() and NE() functions with the RCODE function. Following are the possible values: NOERROR, FORMERR, SERVFAIL, NXDOMAIN, NOTIMP, and REFUSED.
DNS.RES.QUESTION.DOMAIN	Return the domain name (the value of the QNAME field) in the question section of the DNS response. The domain name is returned as a text string, which can be passed to EQ(), NE(), and any other functions that work with text.
DNS.RES.QUESTION.TYPE	Return the query type (the value of the QTYPE field) in the question section of the DNS response. The field indicates the type of resource record (for example, A, NS, or CNAME) that is contained in the response. The returned value can be compared to one of the following values by using the EQ()and NE() functions: A, AAAA, NS, SRV, PTR, CNAME, SOA, MX, and ANY. You can use only the EQ() and NE() functions with the TYPE function. Example: DNS.RES.QUESTION.TYPE.EQ(SOA)

The following functions return the transport layer protocol name.

Function	Description
DNS.REQ.TRANSPORT	Return the name of the transport layer protocol that was used to send the DNS query. Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. Example: DNS.REQ.TRANSPORT.EQ(TCP)

Function	Description
DNS.RES.TRANSPORT	Return the name of the transport layer protocol that was used for the DNS response. Possible values returned are TCP and UDP. You can use only the EQ() and NE() functions with the TRANSPORT function. Example: DNS.RES.TRANSPORT.EQ(TCP)

XPath and HTML, XML, or JSON expressions

September 21, 2020

The Advanced policy infrastructure supports expressions for evaluating and retrieving data from HTML, XML, and JavaScript Object Notation (JSON) files. This enables you to find specific nodes in an HTML, XML, or JSON document, determine if a node exists in the file, locate nodes in XML contexts (for example, nodes that have specific parents or a specific attribute with a given value), and return the contents of such nodes. Additionally, you can use XPath expressions in rewrite expressions.

The Advanced policy expression implementation for XPath comprises an Advanced policy expression prefix (such as "HTTP.REQ.BODY") that designates HTML or XML text, and the XPATH operator that takes the XPath expression as its argument.

HTML files are a largely free-form collection of tags and text elements. You can use the XPATH_HTML operator, which takes an XPath expression as its argument, to process HTML files. JSON files are either a collection of name/value pairs or an ordered list of values. You can use the XPATH_JSON operator, which takes an XPath expression as its argument, to process JSON files.

- **<text>.XPATH(xpathex):**
  Operate on an XML file and return a Boolean value.

  For example, the following expression returns a Boolean TRUE if a node called "creator" exists under the node "Book" within the first 1000 bytes of the XML file.

  ```
  HTTP.REQ.BODY(1000).XPATH(xp%boolean(//Book/creator)%)
  ```

  Parameters:

  xpathex - XPath Boolean expression

- **<text>.XPATH(xpathex):**
  Operate on an XML file and return a value of data type "double."

For example, the following expression converts the string "36" (a price value) to a value of data type "double" if the string is in the first 1000 bytes of the XML file:

`HTTP.REQ.BODY(1000).XPATH(xp%number(/Book/price)%)`

Parameters:

xpathex - XPath numeric expression

- **<text>.XPATH(xpathex):**
  Operate on an XML file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine.

  For example, the following expression selects all the nodes that are enclosed by "/Book/creator" (a node-set) in the first 1000 bytes of the body:

  `HTTP.REQ.BODY(1000).XPATH(xp%/Book/creator%)`

  Parameters:

  xpathex - XPath expression

Example

```
    <Book>
    <creator>
        <Person>
            <name>Milton</name>
        </Person>
    </creator>
    <title>Paradise Lost</title>
    </Book>
<!--NeedCopy-->
```

- **<text>.XPATH_HTML(xpathex)**

  Operate on an HTML file and return a text value.

  For example, the following expression operates on an HTML file and returns the text enclosed in <title></title> tags if the title HTML element is found in the first 1000 bytes:

  `HTTP.REQ.BODY(1000).XPATH_HTML(xp%/html/head/title%)`

  Parameters:

  xpathex - XPath text expression

- **<text>.XPATH_HTML_WITH_MARKUP(xpathex)**

  Operate on an HTML file and return a string that contains the entire selected portion of the document, including markup such as including the enclosing element tags.

The following expression operates on the HTML file and selects all content within the <title> tag, including markup.

`HTTP.REQ.BODY(1000).XPATH_HTML_WITH_MARKUP( xp%/html/head/title%)`

The portion of the HTML body that is selected by the expression is marked for further processing.

Parameters:

xpathex - XPath expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a Boolean value.

  For example, consider the following JSON file:

  { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } }

  The following expression operates on the JSON file and returns a Boolean TRUE if the JSON file contains a node named "creator," whose parent node is "Book," in the first 1000 bytes:

  `HTTP.REQ.BODY(1000).XPATH_JSON(xp%boolean(/Book/creator)%)`

  Parameters:

  xpathex - XPath Boolean expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a value of data type "double."

  For example, consider the following JSON file:

  { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>', "price":"36" } }

  The following expression operates on the JSON file and converts the string "36" to a value of data type "double" if the string is present in the first 1000 bytes of the JSON file.

  `HTTP.REQ.BODY(1000).XPATH_JSON(xp%number(/Book/price)%)`

  Parameters:

  xpathex - XPath numeric expression

- **<text>.XPATH_JSON(xpathex)**

  Operate on a JSON file and return a node-set or a string. Node-sets are converted to corresponding strings by using the standard XPath string conversion routine.

  For example, consider the following JSON file:

  { "Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } }

  The following expression selects all the nodes that are enclosed by "/Book" (a node-set) in the first 1000 bytes of the body of the JSON file and returns the corresponding string value, which is "<name><title>":

`HTTP.REQ.BODY(1000).XPATH_JSON(xp%/Book%)`

Parameters:

xpathex - XPath expression

- **<text>.XPATH_JSON_WITH_MARKUP(xpathex)**
  Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags.

  For example, consider the following JSON file:

  {"Book":{ "creator":{ "person":{ "name":'<name>' } }, "title":'<title>' } }

  The following expression operates on the JSON file and selects all the nodes that are enclosed by "/Book/creator" in the first 1000 bytes of the body, which is "creator:{ person:{ name:'<name>' } }."

  `HTTP.REQ.BODY(1000).XPATH_JSON_WITH_MARKUP(xp%/Book/creator%)`

  The portion of the JSON body that is selected by the expression is marked for further processing.

  Parameters:

  xpathex - XPath expression

- **<text>.XPATH_WITH_MARKUP(xpathex):**
  Operate on an XML file and return a string that contains the entire portion of the document for the result node, including markup such as including the enclosing element tags.

  For example, the following expression operates on an XML file and selects all the nodes enclosed by "/Book/creator" in the first 1000 bytes of the body.

  `HTTP.REQ.BODY(1000).XPATH_WITH_MARKUP(xp%/Book/creator%)`

  The portion of the JSON body that is selected by the expression is marked for further processing.

  Parameters:

  xpathex - XPath expression

Encrypt and decrypt XML payloads

September 21, 2020

You can use the XML_ENCRYPT() and XML_DECRYPT() functions in Advanced policy expressions to encrypt and decrypt, respectively, XML data. These functions conform to the W3C XML Encryption standard defined at "http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/." XML_ENCRYPT() and XML_DECRYPT() support a subset of the XML Encryption specification. In the subset, data encryption uses a bulk cipher method (RC4, DES3, AES128, AES192, or AES256), and an RSA public key is used to encrypt the bulk cipher key.

Note: If you want to encrypt and decrypt text in a payload, you must use the
ENCRYPT and
DECRYPT functions. For more information about these functions, see
Encrypt and decrypt text.

The
XML_ENCRYPT() and
XML_DECRYPT() functions are not dependent on the encryption/decryption service that is used by the
ENCRYPT and
DECRYPT commands for text. The cipher method is specified explicitly as an argument to the
XML_ENCRYPT() function. The
XML_DECRYPT() function obtains the information about the specified cipher method from the
<xenc:EncryptedData> element. Following are synopses of the XML encryption and decryption functions:

- **XML_ENCRYPT(<certKeyName>, <method> [, <flags>])**. Returns an <xenc:EncryptedData> element that contains the encrypted input text and the encryption key, which is itself encrypted by using RSA.
- **XML_DECRYPT(<certKeyName>)**. Returns the decrypted text from the input <xenc:EncryptedData> element, which includes the cipher method and the RSA-encrypted key.

Note: The
<xenc:EncryptedData> element is defined in the W3C XML Encryption specification.

Following are descriptions of the arguments:

- **certKeyName:** Selects an X.509 certificate with an RSA public key for XML_ENCRYPT() or an RSA private key for XML_DECRYPT(). The certificate key must have been previously created by an add ssl certKey command.
- **method:** Specifies which cipher method to use for encrypting the XML data. Possible values: RC4, DES3, AES128, AES192, AES256.
- **flags:** A bitmask specifying the following optional key information (
  <ds:KeyInfo>) to be included in the
  <xenc:EncryptedData> element that is generated by
  XML_ENCRYPT():
  - **1** - Include a KeyName element with the certKeyName. The element is <ds:KeyName>.
  - **2** - Include a KeyValue element with the RSA public key from the certificate. The element is <ds:KeyValue>.
  - **4** - Include an X509IssuerSerial element with the certificate serial number and issuer DN. The element is <ds:X509IssuserSerial>.

  - **8** - Include an X509SubjectName element with the certificate subject DN. The element is <ds:X509SubjectName>.
  - **16** - Include an X509Certificate element with the entire certificate. The element is <ds:X509Certificate>.

Use the XML_ENCRYPT() and XML_DECRYPT() functions in expressions

The XML encryption feature uses SSL certificate-key pairs to provide X.509 certificates (with RSA public keys) for key encryption and RSA private keys for key decryption. Therefore, before you use the XML_ENCRYPT() function in an expression, you must create an SSL certificate-key pair. The following command creates an SSL certificate-key pair, my-certkey, with the X.509 certificate, my-cert.pem, and the private key file, my-key.pem.

`add ssl certKey my-certkey -cert my-cert.pem -key my-key.pem -passcrypt kxPeMRYnitY=`

The following CLI commands create rewrite actions and policies for encrypting and decrypting XML content.

```
add rewrite action my-xml-encrypt-action replace "HTTP.RES.BODY(10000).
    XPATH_WITH_MARKUP(xp%/%)" "HTTP.RES.BODY(10000).XPATH_WITH_MARKUP(xp
    %/%).XML_ENCRYPT("my-certkey", AES256, 31)" -bypassSafetyCheck YES

add rewrite action my-xml-decrypt-action replace "HTTP.REQ.BODY(10000).
    XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)" "HTTP.REQ.BODY(10000).
    XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%).XML_DECRYPT("my-certkey"
    )" -bypassSafetyCheck YES

add rewrite policy my-xml-encrypt-policy "HTTP.REQ.URL.CONTAINS("xml-
    encrypt")" my-xml-encrypt-action

add rewrite policy my-xml-decrypt-policy "HTTP.REQ.BODY(10000).XPATH(xp
    %boolean(//xenc:EncryptedData)%)" my-xml-decrypt-action

bind rewrite global my-xml-encrypt-policy 30

bind rewrite global my-xml-decrypt-policy 30
<!--NeedCopy-->
```

In the above example, the rewrite action my-xml-encrypt-action encrypts the entire XML document (XPATH_WITH_MARKUP(xp%/%)) in the request by using the AES-256 bulk encryption method and the RSA public key from my-certkey to encrypt the bulk encryption key. The action replaces the document with an <xenc:EncryptedData> element containing the encrypted data and an encrypted key. The flags represented by 31 include all of the optional <ds:KeyInfo> elements.

The action my-xml-decrypt-action decrypts the first <xenc:EncryptedData> element in the response (XPATH_WITH_MARKUP(xp%//xenc:EncryptedData%)). This requires the prior addition of the xenc XML namespace by use of the following CLI command:

```
add ns xmlnamespace xenc http://www.w3.org/2001/04/xmlenc##
```

The my-xml-decrypt-action action uses the RSA private key in my-certkey to decrypt the encrypted key and then uses the bulk encryption method specified in the element to decrypt the encrypted contents. Finally, the action replaces the encrypted data element with the decrypted content.

The rewrite policy my-xml-encrypt-policy applies my-xml-encrypt-action to requests for URLs containing xml-encrypt. The action encrypts the entire response from a service configured on the NetScaler appliance.

The rewrite policy my-xml-decrypt-policy applies my-xml-decrypt-action to requests that contain an <xenc:EncryptedData> element ((XPATH(xp%//xenc:EncryptedData%) returns a non-empty string). The action decrypts the encrypted data in requests that are bound for a service configured on the NetScaler appliance.

Advanced policy expressions: parsing SSL

September 21, 2020

There are advanced policy expressions to parse SSL certificates and SSL client hello messages.

Parse SSL certificates

You can use advanced policy expressions to evaluate X.509 Secure Sockets Layer (SSL) client certificates. A client certificate is an electronic document that can be used to authenticate a user's identity. A client certificate contains (at a minimum) version information, a serial number, a signature algorithm ID, an issuer name, a validity period, a subject (user) name, a public key, and signatures.

You can examine both SSL connections and data in client certificates. For example, you may want to send SSL requests that use low-strength ciphers to a particular load balancing virtual server farm. The following command is an example of a Content Switching policy that parses the cipher strength in a request and matches cipher strengths that are less than or equal to 40:

```
add cs policy p1 -rule "client.ssl.cipher\_bits.le(40)"
<!--NeedCopy-->
```

As another example, you can configure a policy that determines whether a request contains a client certificate:

```
add cs policy p2 -rule "client.ssl.client\_cert EXISTS"
<!--NeedCopy-->
```

Or, you might want to configure a policy that examines particular information in a client certificate. For example, the following policy verifies that the certificate has one or more days before expiration:

```
add cs policy p2 -rule "client.ssl.client\_cert exists && client.ssl.
    client\_cert.days\_to\_expire.ge(1)"
<!--NeedCopy-->
```

> **Note**
>
> For information on parsing dates and times in a certificate, see Format of Dates and Times in an Expression" and
> Expressions for SSL Certificate Dates.

Prefixes for text-based SSL and certificate data

The following table describes expression prefixes that identify text-based items in SSL transactions and client certificates.

Table 1. Prefixes That Return Text or Boolean Values for SSL and Client Certificate Data

Prefix	Description
CLIENT.SSL.CLIENT_CERT	Returns the SSL client certificate in the current SSL transaction.
CLIENT.SSL.CLIENT_CERT.TO_PEM	Returns the SSL client certificate in binary format.
CLIENT.SSL.CIPHER_EXPORTABLE	Returns a Boolean TRUE if the SSL cryptographic SSL cryptographic cipher is exportable.
CLIENT.SSL.CIPHER_NAME	Returns the name of the SSL Cipher if invoked from an SSL connection, and a NULL string if invoked from a non-SSL connection.
CLIENT.SSL.IS_SSL	Returns a Boolean TRUE if the current connection is SSL-based.

Prefixes for numeric data in SSL certificates

The following table describes prefixes that evaluate numeric data other than dates in SSL certificates. These prefixes can be used with the operations that are described in Basic Operations on Expression Prefixes" and Compound Operations for Numbers.

Table 2. Prefixes That Evaluate Numeric Data Other Than Dates in SSL Certificates

Prefix	Description
CLIENT.SSL.CLIENT_CERT.DAYS_TO_EXPIRE	Returns the number of days that the certificate is valid, or returns -1 for expired certificates.
CLIENT.SSL.CLIENT_CERT.PK_SIZE	Returns the size of the public key used in the certificate.
CLIENT.SSL.CLIENT_CERT.VERSION	Returns the version number of the certificate. If the connection is not SSL-based, returns zero (0).
CLIENT.SSL.CIPHER_BITS	Returns the number of bits in the cryptograhic key. Returns 0 if the connection is not SSL-based.
CLIENT.SSL.VERSION	Returns a number that represents the SSL protocol version, as follows: 0. The transaction is not SSL-based.; 0x002. The transaction is SSLv2.; 0x300. The transaction is SSLv3.; 0x301. The transaction is TLSv1.; 0x302. The transaction is TLSv1.1.; 0x303. The transaction is TLSv1.2.

> **Note**
>
> For expressions related to expiration dates in a certificate, see Expressions for SSL Certificate Dates.

Expressions for SSL certificates

You can parse SSL certificates by configuring expressions that use the following prefix:

CLIENT.SSL.CLIENT_CERT

This section discusses the expressions that you can configure for certificates, with the exception of expressions that examine certificate expiration. Time-based operations are described in Advanced Policy Expressions: Working with Dates, Times, and Numbers.

The following table describes operations that you can specify for the CLIENT.SSL.CLIENT_CERT prefix.

Table 3. Operations That Can Be Specified with the CLIENT.SSL.CLIENT_CERT Prefix

SSL Certificate Operation	Description
`<certificate>.EXISTS`	Returns a Boolean TRUE if the client has an SSL certificate.
`<certificate>.ISSUER`	Returns the Distinguished Name (DN) of the Issuer in the certificate as a name-value list. An equals sign ("=") is the delimiter for the name and the value, and the slash ("/") is the delimiter that separates the name-value pairs. Following is an example of the returned DN: `/C=US/O=myCompany/OU=www.mycompany.com/CN=www.mycompany.com/emailAddress=myuserid@mycompany.com`
`<certificate>.ISSUER.IGNORE_EMPTY_ELEMENTS`	Returns the Issuer and ignores the empty elements in a name-value list. For example, consider the following: Cert-Issuer: /c=in/st=kar//l=bangelore //o=mycompany/ou=sales/ /emailAddress=myuserid@mycompany.com. The following Rewrite action returns a count of 6 based on the preceding Issuer definition: `sh rewrite action insert_ssl_header Name: insert_sslOperation: insert_http_header Target:Cert-Issuer Value:CLIENT.SSL.CLIENT_CERT.ISSUER.COUNT`. However, if you change the value to the following, the returned count is 9: `CLIENT.SSL.CLIENT_CERT.ISSUER.IGNORE_EMPTY_ELEMENTS.COUNT`

Advanced policy expressions: IP and MAC addresses, throughput, VLAN IDs

September 21, 2020

You can use Advanced policy expression prefixes that return IPv4 and IPv6 addresses, MAC addresses, IP subnets, useful client and server data such as the throughput rates at the interface ports (Rx, Tx, and RxTx), and the IDs of the VLANs through which packets are received. You can then use various operators to evaluate the data that is returned by these expression prefixes.

Expressions for IP Addresses and IP Subnets

You can use Advanced policy expressions to evaluate addresses and subnets that are in Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) format. Expression prefixes for IPv6 addresses and subnets include IPv6 in the prefix. Expression prefixes for IPv4 addresses and subnets include IP in the prefix. Following is an example of an expression that identifies whether a request has originated from a particular IPv4 subnet.

```
client.ip.src.in_subnet(147.1.0.0/16)
<!--NeedCopy-->
```

Following are two examples of Rewrite policies that examine the subnet from which the packet is received and perform a rewrite action on the Host header. With these two policies configured, the rewrite action that is performed depends on the subnet in the request. These two policies evaluate IP addresses that are in the IPv4 address format.

```
- add rewrite action URL1-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany1.com\""
- add rewrite policy URL1-rewrite-policy "http.req.header(\"Host\").contains(\"www.test1.com\") && client.ip.src.in_subnet(147.1.0.0/16)" URL1-rewrite-action
- add rewrite action URL2-rewrite-action replace "http.req.header(\"Host\")" "\"www.mycompany2.com\""
- add rewrite policy URL2-rewrite-policy "http.req.header(\"Host\").contains(\"www.test2.com\") && client.ip.src.in_subnet(10.202.0.0/16)" URL2-rewrite-action
<!--NeedCopy-->
```

> **Note**
>
> The preceding examples are commands that you type at the NetScaler command-line interface (CLI) and, therefore, each quotation mark must be preceded by a backslash (\). For more infor-

> mation, see Configuring advanced policy expressions in a policy.

Prefixes for IPV4 Addresses and IP Subnets

The following table describes prefixes that return IPv4 addresses and subnets, and segments of IPv4 addresses. You can use numeric operators and operators that are specific to IPv4 addresses with these prefixes. For more information about numeric operations, see Basic Operations on Expression Prefixes and Compound Operations for Numbers.

Table 1. Prefixes That Evaluate IP and MAC Addresses

Prefix	Description
CLIENT.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
CLIENT.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.
SERVER.IP.SRC	Returns the source IP of the current packet as an IP address or as a number.
SERVER.IP.DST	Returns the destination IP of the current packet as an IP address or as a number.

Operations for IPV4 Addresses

The following table describes the operators that can be used with prefixes that return an IPv4 address.

Prefix	Description
`<ip address>.EQ(<address>)`	Returns a Boolean TRUE if the IP address value is same as the `<address>` argument. The following example checks whether the client's destination IP address is equal to 10.100.10.100: `client.ip.dst.eq(10.100.10.100)`
`<ip address>.GET1. . .GET4`	Returns a portion of an IP address as a numeric value. For example, if the IP address value is 10.100.200.1, the following is returned: `client.ip.src.get1 Returns 10;` `client.ip.src.get2 returns 100;` `client.ip.src.get3 returns 200`

Prefix	Description
`<ip address>.IN_SUBNET(<subnet>)`	Returns a Boolean TRUE if the `<subnet>` argument matches the subnet of the IP address value. For example, the following determines whether the client's destination IP address subnet is 10.100.10.100/18: `client.ip.dst.eq(10.100.10.100/18)`
`<ip address>.SUBNET(<n>)`	Returns the IP address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 32. For example: `CLIENT.IP.SRC.SUBNET (24)returns 192.168.1.0 if the IP address represented by the prefix is 192.168.1.[0-255].`
`<ip address>.IS_IPV6`	Returns a Boolean TRUE if this is an Internet Protocol version 6 (IPv6) host for the client or server. Following is an example: `client.ip.src.is_ipv6`
`<ip address>.MATCHES(<hostname>)`	Returns a Boolean TRUE if the IP address for the host specified in `<hostname>` matches the current IP address. The `<hostname>` cannot exceed 255 characters.
`<ip address>.MATCHES_LOCATION(< location>)`	Returns a Boolean TRUE if the location of the IP address matches the `<location>` argument. The Location string can take the following form: qual1.qual2.qual3.qual4.qual5.qual6, `for example: NorthAmeria.CA.* Following is an example: client.ip. src.matches_location \"Europe.GB .17.London.*.*\")`

About IPv6 expressions

The IPv6 address format allows more flexibility than the older IPv4 format. IPv6 addresses are in the hexadecimal format (RFC 2373). In the following examples, Example 1 is an IPv6 address, Example 2 is a URL that includes the IPv6 address, and Example 3 includes the IPv6 address and a port number.

Example 1:

```
9901:0ab1:22a2:88a3:3333:4a4b:5555:6666
<!--NeedCopy-->
```

Example 2:

```
http://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]/
<!--NeedCopy-->
```

Example 3:

```
https://[9901:0ab1:22a2:88a3:3333:4a4b:5555:6666]:8080/
<!--NeedCopy-->
```

In Example 3, the brackets separate the IP address from the port number (8080).

Note that you can only use the '+' operator to combine IPv6 expressions with other expressions. The output is a concatenation of the string values that are returned from the individual expressions. You cannot use any other arithmetic operator with an IPv6 expression. The following syntax is an example:

```
client.ipv6.src + server.ip.dst
<!--NeedCopy-->
```

For example, if the client source IPv6 address is ABCD:1234::ABCD, and the server destination IPv4 address is 10.100.10.100, the preceding expression returns "ABCD:1234::ABCD10.100.10.100".

Note that when the NetScaler appliance receives an IPv6 packet, it assigns a temporary IPv4 address from an unused IPv4 address range and changes the source address of the packet to this temporary address. At response time, the outgoing packet's source address is replaced with the original IPv6 address.

> **Note**
>
> You can combine an IPv6 expression with any other expression except an expression that produces a Boolean result.

Expression prefixes for IPv6 addresses

The IPv6 addresses that are returned by the expression prefixes in the following table can be treated as text data. For example, the prefix client.ipv6.dst returns the destination IPv6 address as a string that can be evaluated as text.

The following table describes expression prefixes that return an IPv6 address.

Table 3. IPv6 Expression Prefixes That Return Text

Prefix	Description
CLIENT.IPV6	Operates on the IPv6 address in with the current packet.
CLIENT.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
CLIENT.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: `client.ipv6.src.in_subnet(2007::2008/64)` `client.ipv6.src.get1.le(2008)`
SERVER.IPV6	Operates on the IPv6 address in with the current packet.
SERVER.IPV6.DST	Returns the IPv6 address in the destination field of the IP header.
SERVER.IPV6.SRC	Returns the IPv6 address in the source field of the IP header. Following are examples: `server.ipv6.src.in_subnet(2007::2008/64)` `server.ipv6.src.get1.le(2008)`

Operations for IPv6 prefixes

The following table describes the operators that can be used with prefixes that return an IPv6 address:

Table 4. Operations That Evaluate IPv6 Addresses

IPv6 Operation	Description
`<ipv6>.EQ(<IPv6_address>)`	Returns a Boolean TRUE if the IP address value is same as the argument. Following is an example: ‘client.ipv6.dst.eq(ABCD:1234::ABCD)‘

IPv6 Operation	Description
`<ipv6>.GET1. . .GET8`	Returns a segment of an IPv6 address as a number. The following example expressions retrieve segments from the ipv6 address 1000:1001:CD10:0000:0000:89AB:4567:CDEF: `client.ipv6.dst.get5 extracts 0000`, which is the fifth set of bits in the address. `client.ipv6.dst.get6 extracts 89AB`. `client.ipv6.dst.get7 extracts 4567`. You can perform numeric operations on these segments. Note that you cannot perform numeric operations when you retrieve an entire IPv6 address. This is because expressions that return an entire IPv6 address, such as CLIENT.IPV6.SRC, return the address in text format.
`<ipv6>.IN_SUBNET(<subnet>)`	Returns a Boolean TRUE if the IPv6 address value is in the subnet specified by the `<subnet>` argument. Following is an example: `client.ipv6.dst.eq(1000:1001:CD10 :0000:0000:89AB:4567:CDEF/60)`
`<ipv6>.IS_IPV4`	Returns a Boolean TRUE if this is an IPv4 client, and returns a Boolean FALSE if it is not.
`<ipv6>.SUBNET(<n>)`	Returns the IPv6 address after applying the subnet mask specified as the argument. The subnet mask can take values between 0 and 128. For example: `CLIENT.IPV6.SRC.SUBNET(24)`

Expressions for MAC addresses

A MAC address consists of colon-delimited hexadecimal values in the format ##:##:##:##:##:##, where each “#” represents either a number from 0 through 9 or a letter from A through F. Default syntax expression prefixes and operators are available for evaluating source and destination MAC addresses.

Prefixes for MAC addresses

The following table describes prefixes that return MAC addresses.

Table 5. Prefixes That Evaluate MAC Addresses

Prefix	Description
client.ether.dstmac	Returns the MAC address in the destination field of the Ethernet header.
client.ether.srcmac	Returns the MAC address in the source field of the Ethernet header.

Operations for MAC addresses

The following table describes the operators that can be used with prefixes that return a MAC address.

Table 6. Operations on MAC Addresses

Prefix	Description
`<mac address>.EQ(<address>)`	Returns a Boolean TRUE if the MAC address value is same as the `<address>` argument.
`<mac address>.GET1. . .GET4`	Returns a numeric value extracted from the segment of the MAC address that is specified in the GET operation. For example, if the MAC address is 12:34:56:78:9a:bc, the following returns 34: `client.ether.dstmac.get2`

Expressions for Numeric Client and Server Data

The following table describes prefixes for working with numeric client and server data, including throughput, port numbers, and VLAN IDs.

Table 7. Prefixes That Evaluate Numeric Client and Server Data

Prefix	Description
client.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in kilobytes per second (KBps) for the previous seven seconds.
client.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.

Prefix	Description
client.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxthroughput	Returns an integer representing the raw received traffic throughput in KBps for the previous seven seconds.
server.interface.txthroughput	Returns an integer representing the raw transmitted traffic throughput in KBps for the previous seven seconds.
server.interface.rxtxthroughput	Returns an integer representing the raw received and transmitted traffic throughput in KBps for the previous seven seconds.
server.vlan.id	Returns a numeric ID of the VLAN through which the current packet entered the NetScaler.
client.vlan.id	Returns a numeric ID for the VLAN through which the current packet entered the NetScaler.

Advanced policy expressions: stream analytics functions

September 21, 2020

Stream Analytics expressions begin with the ANALYTICS.STREAM(<identifier_name>) prefix. The following list describes the functions that can be used with this prefix.

- **COLLECT_STATS**

  Collect statistical data from the requests that are evaluated against the policy and create a record for each request.

- **REQUESTS**

  Return the number of requests that exist for the specified record grouping. The value returned is of type unsigned long.

- **BANDWIDTH**

Return the bandwidth statistic for the specified record grouping. The value returned is of type unsigned long.

- **RESPTIME**

  Return the response time statistic for the specified record grouping. The value returned is of type unsigned long.

- **CONNECTIONS**

  Return the number of concurrent connections that exist for the specified record grouping. The value returned is of type unsigned long.

- **IS_TOP(n)**

  Return a Boolean
  TRUE if the statistical value for the specified record grouping is one among the top
  n groups. Otherwise, return a Boolean
  FALSE.

- **CHECK_LIMIT**

  Return a Boolean
  TRUE if the statistic for the specified record grouping has hit the preconfigured limit. Otherwise, return a Boolean
  FALSE.

Advanced policy expressions: DataStream

September 21, 2020

The policy infrastructure on the NetScaler appliance includes expressions that you can use to evaluate and process database server traffic when the appliance is deployed between a farm of application servers and their associated database servers.

This topic includes the following sections:

- Expressions for the MySQL Protocol
- Expressions for Evaluating Microsoft SQL Server Connections

Expressions for the MySQL protocol

The following expressions evaluate traffic associated with MySQL database servers. You can use the request-based expressions (expressions that begin with MYSQL.CLIENT and MYSQL.REQ) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

- **MYSQL.CLIENT.** Operates on the client properties of a MySQL connection.
- **MYSQL.CLIENT.CAPABILITIES.** Returns the set of flags that the client has set in the capabilities field of the handshake initialization packet during authentication. Examples of the flags that are set are CLIENT_FOUND_ROWS, CLIENT_COMPRESS, and CLIENT_SSL.
- **MYSQL.CLIENT.CHAR_SET.** Returns the enumeration constant assigned to the character set that the client uses. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the character set enumeration constants:
    - LATIN2_CZECH_CS
    - DEC8_SWEDISH_CI
    - CP850_GENERAL_CI
    - GREEK_GENERAL_CI
    - LATIN1_GERMAN1_CI
    - HP8_ENGLISH_CI
    - KOI8R_GENERAL_CI
    - LATIN1_SWEDISH_CI
    - LATIN2_GENERAL_CI
    - SWE7_SWEDISH_CI
    - ASCII_GENERAL_CI
    - CP1251_BULGARIAN_CI
    - LATIN1_DANISH_CI
    - HEBREW_GENERAL_CI
    - LATIN7_ESTONIAN_CS
    - LATIN2_HUNGARIAN_CI
    - KOI8U_GENERAL_CI
    - CP1251_UKRAINIAN_CI
    - CP1250_GENERAL_CI
    - LATIN2_CROATIAN_CI
    - CP1257_LITHUANIAN_CI
    - LATIN5_TURKISH_CI
    - LATIN1_GERMAN2_CI
    - ARMSCII8_GENERAL_CI
    - UTF8_GENERAL_CI
    - CP1250_CZECH_CS
    - CP866_GENERAL_CI

  - KEYBCS2_GENERAL_CI
  - MACCE_GENERAL_CI
  - MACROMAN_GENERAL_CI
  - CP852_GENERAL_CI
  - LATIN7_GENERAL_CI
  - LATIN7_GENERAL_CS
  - MACCE_BIN
  - CP1250_CROATIAN_CI
  - LATIN1_BIN
  - LATIN1_GENERAL_CI
  - LATIN1_GENERAL_CS
  - CP1251_BIN
  - CP1251_GENERAL_CI
  - CP1251_GENERAL_CS
  - MACROMAN_BIN
  - CP1256_GENERAL_CI
  - CP1257_BIN
  - CP1257_GENERAL_CI
  - ARMSCII8_BIN
  - ASCII_BIN
  - CP1250_BIN
  - CP1256_BIN
  - CP866_BIN
  - DEC8_BIN
  - GREEK_BIN
  - HEBREW_BIN
  - HP8_BIN
  - KEYBCS2_BIN
  - KOI8R_BIN
  - KOI8U_BIN
  - LATIN2_BIN
  - LATIN5_BIN
  - LATIN7_BIN
  - CP850_BIN
  - CP852_BIN
  - SWE7_BIN
  - UTF8_BIN
  - GEOSTD8_GENERAL_CI
  - GEOSTD8_BIN

  - LATIN1_SPANISH_CI
  - UTF8_UNICODE_CI
  - UTF8_ICELANDIC_CI
  - UTF8_LATVIAN_CI
  - UTF8_ROMANIAN_CI
  - UTF8_SLOVENIAN_CI
  - UTF8_POLISH_CI
  - UTF8_ESTONIAN_CI
  - UTF8_SPANISH_CI
  - UTF8_SWEDISH_CI
  - UTF8_TURKISH_CI
  - UTF8_CZECH_CI
  - UTF8_DANISH_CI
  - UTF8_LITHUANIAN_CI
  - UTF8_SLOVAK_CI
  - UTF8_SPANISH2_CI
  - UTF8_ROMAN_CI
  - UTF8_PERSIAN_CI
  - UTF8_ESPERANTO_CI
  - UTF8_HUNGARIAN_CI
  - INVAL_CHARSET

- **MYSQL.CLIENT.DATABASE.** Returns the name of the database specified in the authentication packet that the client sends to the database server. This is the databasename attribute.
- MYSQL.CLIENT.USER. Returns the user name (in the authentication packet) with which the client is attempting to connect to the database. This is the user attribute.
- **MYSQL.REQ.** Operates on a MySQL request.
- **MYSQL.REQ.COMMAND.** Identifies the enumeration constant assigned to the type of command in the request. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. Following are the enumeration constant values:
  - SLEEP
  - QUIT
  - INIT_DB
  - QUERY
  - FIELD_LIST
  - CREATE_DB
  - DROP_DB
  - REFRESH

  - SHUTDOWN
  - STATISTICS
  - PROCESS_INFO
  - CONNECT
  - PROCESS_KILL
  - DEBUG
  - PING
  - TIME
  - DELAYED_INSERT
  - CHANGE_USER
  - BINLOG_DUMP
  - TABLE_DUMP
  - CONNECT_OUT
  - REGISTER_SLAVE
  - STMT_PREPARE
  - STMT_EXECUTE
  - STMT_SEND_LONG_DATA
  - STMT_CLOSE
  - STMT_RESET
  - SET_OPTION
  - STMT_FETCH

- **MYSQL.REQ.QUERY.** Identifies the query in the MySQL request.
- **MYSQL.REQ.QUERY.COMMAND.** Returns the first keyword in the MySQL query.
- **MYSQL.REQ.QUERY.SIZE.** Returns the size of the request query in integer format. The SIZE method is similar to the CONTENT_LENGTH method that returns the length of an HTTP request or response.
- **MYSQL.REQ.QUERY.TEXT.** Returns a string covering the entire query.
- **MYSQL.REQ.QUERY.TEXT(<n>).** Returns the first n bytes of the MySQL query as a string. This is similar to HTTP.BODY(<n>).

  Parameters:

  n - Number of bytes to be returned

- **MYSQL.RES.** Operates on a MySQL response.
- **MYSQL.RES.ATLEAST_ROWS_COUNT(<i>).** Checks whether the response has at least i number of rows and returns a Boolean TRUE or FALSE to indicate the result.

  Parameters:

  i - Number of rows

- **MYSQL.RES.ERROR.** Identifies the MySQL error object. The error object includes the error number and the error message.
- **MYSQL.RES.ERROR.MESSAGE.** Returns the error message that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.NUM.** Returns the error number that is retrieved from the server's error response.
- **MYSQL.RES.ERROR.SQLSTATE.** Returns the value of the SQLSTATE field in the server's error response. The MySQL server translates error number values to SQLSTATE values.
- **MYSQL.RES.FIELD(<i>).** Identifies the packet that corresponds to the i<sup>th</sup> individual field in the server's response. Each field packet describes the properties of the associated column. The packet count (i) begins at 0.

  Parameters:

  i - Packet number
- **MYSQL.RES.FIELD(<i>).CATALOG.** Returns the catalog property of the field packet.
- **MYSQL.RES.FIELD(<i>).CHAR_SET.** Returns the character set of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.
- **MYSQL.RES.FIELD(<i>).DATATYPE.** Returns an enumeration constant that represents the data type of the column. This is the type (also called enum_field_type) attribute of the column. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix. The possible values for the various data types are:
  - DECIMAL
  - TINY
  - SHORT
  - LONG
  - FLOAT
  - DOUBLE
  - NULL
  - TIMESTAMP
  - LONGLONG
  - INT24
  - DATE
  - TIME
  - DATETIME
  - YEAR
  - NEWDATE

  - VARCHAR (new in MySQL 5.0)
  - BIT (new in MySQL 5.0)
  - NEWDECIMAL (new in MySQL 5.0)
  - ENUM
  - SET
  - TINY_BLOB
  - MEDIUM_BLOB
  - LONG_BLOB
  - BLOB
  - VAR_STRING
  - STRING
  - GEOMETRY
- **MYSQL.RES.FIELD(<i>).DB.** Returns the database identifier (db) attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).DECIMALS.** Returns the number of positions after the decimal point if the type is DECIMAL or NUMERIC. This is the decimals attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).FLAGS.** Returns the flags property of the field packet. Following are the possible hexadecimal flag values:
  - 0001: NOT_NULL_FLAG
  - 0002: PRI_KEY_FLAG
  - 0004: UNIQUE_KEY_FLAG
  - 0008: MULTIPLE_KEY_FLAG
  - 0010: BLOB_FLAG
  - 0020: UNSIGNED_FLAG
  - 0040: ZEROFILL_FLAG
  - 0080: BINARY_FLAG
  - 0100: ENUM_FLAG
  - 0200: AUTO_INCREMENT_FLAG
  - 0400: TIMESTAMP_FLAG
  - 0800: SET_FLAG
- **MYSQL.RES.FIELD(<i>).LENGTH.** Returns the length of the column. This is the value of the length attribute of the field packet. The value that is returned might be larger than the actual value. For example, an instance of a VARCHAR(2) column might return a value of 2 even when it contains only one character.
- **MYSQL.RES.FIELD(<i>).NAME.** Returns the column identifier (the name after the AS clause, if any). This is the name attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).ORIGINAL_NAME.** Returns the original column identifier (before the AS clause, if any). This is the org_name attribute of the field packet.

- **MYSQL.RES.FIELD(<i>).ORIGINAL_TABLE.** Returns the original table identifier of the column (before the AS clause, if any). This is the org_table attribute of the field packet.
- **MYSQL.RES.FIELD(<i>).TABLE.** Returns the table identifier of the column (after the AS clause, if any). This is the table attribute of the field packet.
- **MYSQL.RES.FIELDS_COUNT.** Returns the number of field packets in the response (the field_count attribute of the OK packet).
- **MYSQL.RES.OK.** Identifies the OK packet sent by the database server.
- **MYSQL.RES.OK.AFFECTED_ROWS.** Returns the number of rows affected by an INSERT, UPDATE, or DELETE query. This is the value of the affected_rows attribute of the OK packet.
- **MYSQL.RES.OK.INSERT_ID.** Identifies the unique_id attribute of the OK packet. If an auto-increment identity is not generated by the current MySQL statement or query, the value of unique_id, and hence the value returned by the expression, is 0.
- **MYSQL.RES.OK.MESSAGE.** Returns the message property of the OK packet.
- **MYSQL.RES.OK.STATUS.** Identifies the bit string in the server_status attribute of the OK packet. Clients can use the server status to check whether the current command is a part of a running transaction. The bits in the server_status bit string correspond to the following fields (in the given order):
    - IN TRANSACTION
    - AUTO_COMMIT
    - MORE RESULTS
    - MULTI QUERY
    - BAD INDEX USED
    - NO INDEX USED
    - CURSOR EXISTS
    - LAST ROW SEEN
    - DATABASE DROPPED
    - NO BACKSLASH ESCAPES
- **MYSQL.RES.OK.WARNING_COUNT.** Returns the warning_count attribute of the OK packet.
- **MYSQL.RES.ROW(<i>).** Identifies the packet that corresponds to the i<sup>th</sup> individual row in the database server's response.

  Parameters:

  i - Row number
- **MYSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>).** Checks whether the j<sup>th</sup> column of the i<sup>th</sup> row of the table is NULL. Following C conventions, both indexes i and j

start from 0. Therefore, row i and column j are actually the (i+1)<sup>th</sup> row and the (j+1)<sup>th</sup> column, respectively.

Parameters:

i - Row number

j - Column number

- **MYSQL.RES.ROW(<i>).IS_NULL_ELEM(j).** Checks whether the j<sup>th</sup> column of the i<sup>th</sup> row of the table is NULL. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)<sup>th</sup> row and the (j+1)<sup>th</sup> column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.ROW(<i>).NUM_ELEM(<j>).** Returns an integer value from the j<sup>th</sup> column of the i<sup>th</sup> row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)<sup>th</sup> row and the (j+1)<sup>th</sup> column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.ROW(<i>).TEXT_ELEM(j).** Returns a string from the j<sup>th</sup> column of the i<sup>th</sup> row of the table. Following C conventions, both indexes i and j start from 0. Therefore, row i and column j are actually the (i+1)<sup>th</sup> row and the (j+1)<sup>th</sup> column, respectively.

  Parameters:

  i - Row number

  j - Column number

- **MYSQL.RES.TYPE.** Returns an enumeration constant for the response type. Its values can be ERROR, OK, and RESULT_SET. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this prefix.

Expressions for evaluating Microsoft SQL server connections

The following expressions evaluate traffic associated with Microsoft SQL Server database servers. You can use the request-based expressions (expressions that begin with MSSQL.CLIENT and MSSQL.REQ) in policies to make request switching decisions at the content switching virtual server bind point and the response-based expressions (expressions that begin with MSSQL.RES) to evaluate server responses to user-configured health monitors.

Expression	Description
MSSQL.CLIENT.CAPABILITIES	Returns the OptionFlags1, OptionFlags2, OptionFlags3, and TypeFlags fields of the LOGIN7authentication packet, in that order, as a 4-byte integer. Each field is 1 byte long and specifies a set of client capabilities.
MSSQL.CLIENT.DATABASE	Returns the name of the client database. The value returned is of type text.
MSSQL.CLIENT.USER	Returns the user name with which the client authenticated. The value returned is of type text.
MSSQL.REQ.COMMAND	Returns an enumeration constant that identifies the type of command in the request sent to a Microsoft SQL Server database server. The value returned is of type text. Examples of the values of the enumeration constant are QUERY, RESPONSE, RPC, and ATTENTION. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.
MSSQL.REQ.QUERY.COMMAND	Returns the first keyword in the SQL query. The value returned is of type text.
MSSQL.REQ.QUERY.SIZE	Returns the size of the SQL query in the request. The value returned is a number.
MSSQL.REQ.QUERY.TEXT	Returns the entire SQL query as a string. The value returned is of type text.
MSSQL.REQ.QUERY.TEXT(<n>)	Returns the first n bytes of the SQL query. The value returned is of type text. Parameters: n - Number of bytes
MSSQL.REQ.RPC.NAME	Returns the name of the procedure that is being called in a remote procedure call (RPC) request. The name is returned as a string.

Expression	Description
MSSQL.REQ.RPC.IS_PROCID	Returns a Boolean value that indicates whether the remote procedure call (RPC) request contains a procedure ID or an RPC name. A return value of TRUEindicates that the request contains a procedure ID and a return value of FALSE indicates that the request contains an RPC name.
MSSQL.REQ.RPC.PROCID	Returns the procedure ID of the remote procedure call (RPC) request as an integer.
MSSQL.REQ.RPC.BODY	
Note: Not available for releases before 10.1.	Returns the body of the SQL request as a string in the form of parameters represented as "a=b" clauses separated by commas, where "a" is the RPC parameter name and "b" is its value.
MSSQL.REQ.RPC.BODY(n)	
Note: Not available for releases before 10.1.	Returns part of the body of the SQL request as a string in the form of parameters represented as "a=b" clauses separated by commas, where "a" is the RPC parameter name and "b" is its value. Parameters are returned from only the first "n" bytes of the request, skipping the SQL header. Only complete name-value pairs are returned.
MSSQL.RES.ATLEAST_ROWS_COUNT(i)	Checks whether the response has at least i number of rows. The value returned is a Boolean TRUE or FALSEvalue. Parameters: i - Number of rows
MSSQL.RES.DONE.ROWCOUNT	Returns a count of the number of rows affected by an INSERT, UPDATE, or DELETE query. The value returned is of type unsigned long.
MSSQL.RES.DONE.STATUS	Returns the status field from the DONE token sent by a Microsoft SQL Server database server. The value returned is a number.

Expression	Description
MSSQL.RES.ERROR.MESSAGE	Returns the error message from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the MsgText field in the ERROR token. The value returned is of type text.
MSSQL.RES.ERROR.NUM	Returns the error number from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the Number field in the ERROR token. The value returned is a number.
MSSQL.RES.ERROR.STATE	Returns the error state from the ERROR token sent by a Microsoft SQL Server database server. This is the value of the State field in the ERROR token. The value returned is a number.
MSSQL.RES.FIELD(<i>).DATATYPE	Returns the data type of the ith field in the server response. The EQ(<m>) and NE(<m>) functions, which return Boolean values to indicate the result of a comparison, are used with this prefix. For example, the following expression returns a Boolean TRUE if the DATATYPE function returns a value of datetime for the third field in the response: MSSQL.RES.FIELD(<2>).DATATYPE.EQ(datetime) Parameters: i - Row number
MSSQL.RES.FIELD(<i>).LENGTH	Returns the maximum possible length of the ith field in the server response. The value returned is a number. Parameters: i - Row number
MSSQL.RES.FIELD(<i>).NAME	Returns the name of the ith field in the server response. The value returned is of type text. Parameters: i - Row number

Expression	Description
MSSQL.RES.ROW(<i>).DOUBLE_ELEM(<j>)	Returns a value of type double from the jth column of the ith row of the table. If the value is not a double value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.ROW(<i>).NUM_ELEM(j)	Returns an integer value from the jth column of ith row of the table. If the value is not an integer value, an UNDEF condition is raised. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.ROW(<i>).IS_NULL_ELEM(j)	Checks whether the jth column of the ith row of the table is NULL and returns a Boolean TRUE or FALSE to indicate the result. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row iand column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number
MSSQL.RES.ROW(<i>).TEXT_ELEM(j)	Returns a text string from the jth column of ith row of the table. Following C conventions, both indexes i and j start from 0 (zero). Therefore, row i and column j are actually the (i + 1)th row and the (j + 1)th column, respectively. Parameters: i - Row number j - Column number

Expression	Description
MSSQL.RES.TYPE	Returns an enumeration constant that identifies the response type. Following are the possible return values: ERROR, OK, and RESULT_SET. The EQ(<m>) and NE(<m>) operators, which return Boolean values to indicate the result of a comparison, are used with this expression.

Typecasting data

September 21, 2020

You can extract data of one type (for example, text or an integer) from requests and responses and transform it to data of another type. For example, you can extract a string and transform the string to time format. You can also extract a string from an HTTP request body and treat it like an HTTP header or extract a value from one type of request header and insert it in a response header of a different type.

After typecasting the data, you can apply any operation that is appropriate for the new data type. For example, if you typecast text to an HTTP header, you can apply any operation that is applicable to HTTP headers to the returned value.

For more information about typecasting data, see the Typecasting Operations pdf.

Regular Expressions

September 21, 2020

When you want to perform string matching operations that are more complex than the operations that you perform with the `CONTAINS( “\<string\>” )` or EQ(“<string>“) operators, you use regular expressions. The policy infrastructure on the Citrix® NetScaler® appliance includes operators to which you can pass regular expressions as arguments for text matching. The names of the operators that work with regular expressions include the string REGEX. The regular expressions that you pass as arguments must conform to the regular expression syntax that is described in “http://www.pcre.org/pcre.txt.” You can learn more about regular expressions at “http://www.regular-expressions.info/quickstart.html” and at “http://www.silverstones.com/thebat/Regex.html.”

The target text for an operator that works with regular expressions can be either text or the value of an HTTP header. Following is the format of a default syntax expression that uses a regular expression operator to operate on text:

```
<text>.<regex_operator>(re<delimiter><regex_pattern><delimiter>)
```

The string `<text>` represents the default syntax expression prefix that identifies a text string in a packet (for example, HTTP.REQ.URL). The string represents the regular expression operator. The regular expression always begins with the string re. A pair of matching delimiters, represented by “, enclose the string “, which represents the regular expression.

The following example expression checks whether the URL in an HTTP packet contains the string `*.jpeg` (where * is a wildcard) and returns a Boolean TRUE or FALSE to indicate the result. The regular expression is enclosed within a pair of slash marks (/), which act as delimiters.

```
http.req.url.regex_match(re/.<asterisk>\.jpeg/)
```

Regular expression operators can be combined to define or refine the scope of a search. For example, `<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2` specifies that the target for string matching is the text between the patterns `regex_pattern1 and regex_pattern2`. You can use a text operator on the scope that is defined by the regular expression operators. For example, you can use the `CONTAINS( “<string>” )` operator to check whether the defined scope contains the string abc:

```
<text>.AFTER_REGEX(re/regex_pattern1).BEFORE_REGEX(re/regex_pattern2/).
CONTAINS( “abc” )
```

> **Note**
>
> The process of evaluating a regular expression inherently takes more time than that for an operator such as `CONTAINS( “\<string\>” )` or `EQ( “\<string\>” )`, which work with simple string arguments. You should use regular expressions only if your requirement is beyond the scope of other operators.

Basic characteristics of regular expressions

September 21, 2020

Following are notable characteristics of regular expressions as defined on the NetScaler appliance:

- A regular expression always begins with the string “re” followed by a pair of delimiting characters (called delimiters) that enclose the regular expression that you want to use.

  For example, re#<regex_pattern># uses the number sign (#) as a delimiter.

- A regular expression cannot exceed 1499 characters.

- Digit matching can be done by using the string \d (a backslash followed by d).
- White space can be represented by using \s (a backslash followed by s).
- A regular expression can contain white spaces.

Following are the differences between the NetScaler syntax and the PCRE syntax:

- The NetScaler does not allow back references in regular expressions.
- You should not use recursive regular expressions.
- The dot meta-character also matches the newline character.
- Unicode is not supported.
- The operation SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option in the regular expression.

Operations for regular expressions

September 21, 2020

The following table describes the operators that work with regular expressions. The operation performed by a regular expression operator in a given default syntax expression depends on whether the expression prefix identifies text or HTTP headers. Operations that evaluate headers override any text-based operations for all instances of the specified header type. When you use an operator, replace <text> with the default syntax expression prefix that you want to configure for identifying text.

Regular Expression Operation	Description
<text>.BEFORE_REGEX(<regular expression>)	Selects the text that precedes the string that matches the <regular expression> argument. If the regular expression does not match any data in the target, the expression returns a text object of length 0. The following expression selects the string “text” from “text/plain”. http.res.header(“content-type”).before_regex(re#/#)
<text>.AFTER_REGEX(<regular expression>)	Selects the text that follows the string that matches the <regular expression> argument. If the regular expression does not match any text in the target, the expression returns a text object of length 0. The following expression extracts “Example” from “myExample”: http.req.header(“etag”).after_regex(re/my/)

Regular Expression Operation	Description
<text>.REGEX_SELECT(<regular expression>)	Selects a string that matches the <regular expression> argument. If the regular expression does not match the target, a text object of length 0 is returned. The following example extracts the string "NS-CACHE-9.0: 90" from a Via header: http.req.header("via").regex_select(re!NS-CACHE-\d.\d:\s*\d{1,3}!)

Regular Expression Operation	Description
<text>.REGEX_MATCH(<regular expression>)	Returns TRUE if the target matches a <regular expression> argument of up to 1499 characters. The regular expression must be of the following format: re<delimiter>regular expression< delimiter> Both delimiters must be the same. Additionally, the regular expression must conform to the Perl-compatible (PCRE) regular expression library syntax. For more information, go to http://www.pcre.org/pcre.txt. In particular, see the pcrepattern man page. However, note the following: Back-references are not allowed. Recursive regular expressions are not recommended. The dot metacharacter also matches the newline character. The Unicode character set is not supported. SET_TEXT_MODE(IGNORECASE) overrides the (?i) internal option specified in the regular expression. The following are examples: http.req.hostname.regex_match(re/[[:alpha:]]+(abc){2,3}/) and http.req.url.set_text_mode(urlencoded).regex_match(re#(a*b* The following example matches ab and aB: http.req.url.regex_match(re/a(?i)b/) The following example matches ab, aB, Ab and AB: http.req.url.set_text_mode(ignorecase).regex_match(re/a following example performs a case-insensitive, multiline match in which the dot meta-character also matches a newline character: http.req.body.regex_match(re/(?ixm) (^ab (.*) cd$) /)

Configuring classic policies and expressions

September 21, 2020

Some NetScaler features use classic policies and classic expressions. As with default syntax policies, classic policies can be either global or specific to a virtual server. However, to a certain extent, the configuration method and bind points for classic policies are different from those of default syntax policies. As with default syntax expressions, you can configure named expressions and use a named expression in multiple classic policies.

The following table summarizes NetScaler features that can be configured by using classic policies.

Click here to view the table.

Configure a classic policy

September 21, 2020

You can configure classic policies and classic expressions by using either the GUI or the command-line interface. A policy rule cannot exceed 1,499 characters. When configuring the policy rule, you can use named classic expressions. For more information about named expressions, see Create named classic expressions. After configuring the policy, you bind it either globally or to a virtual server.

Note that there are small variations in the policy configuration methods for various NetScaler features.

Note: You can embed a classic expression in a default syntax expression by using the syntax SYS.EVAL_CLASSIC_EXPR(classic_expression), specifying the classic_expression as the argument.

Create a classic policy by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-   add cmp policy <name> -rule <expression> -action <action>

-   show cmp policy [<policyName>]
<!--NeedCopy-->
```

Example

The following commands first create a compression action and then create a compression policy that applies the action:

```
> add cmp action cmp-act-compress compress
 Done
> show cmp action cmp-act-compress
```

```
1)      Name: cmp-act-compress  Compression Type: compress
 Done
> add cmp pol cmp-pol-compress -rule ExpCheckIp -resAction cmp-act-
   compress
 Done
> show cmp pol cmp-pol-compress
1)      Name: cmp-pol-compress  Rule: ExpCheckIp
        Response action: cmp-act-compress      Hits: 0
 Done
>
<!--NeedCopy-->
```

Create a policy with classic expressions by using the GUI

1. In the navigation pane, expand the feature for which you want to configure a policy and, depending on the feature, do the following:
    - For Content Switching, Cache Redirection, and the application firewall, click Policies.
    - For SSL, click Policies, and then in the details pane, click the Policies tab.
    - For System Authentication, click Authentication, and then in the details pane, click the Policies tab.
    - For Filter, SureConnect, and Priority Queuing, expand Protection Features, select the desired function, and then in the details pane, click the Policies tab.
    - For the NetScaler Gateway, expand NetScaler Gateway, expand Policies, select the desired function, and then in the details pane, click the Policies tab.
2. For most features, click the Add button.
3. In the Create <feature name> Policy dialog box, in the Name* text box, enter a name for the policy.

    Note: You must begin a policy name with a letter or underscore. A policy name can consist of 1 to 31 characters, including letters, numbers, hyphen (-), period (.), pound sign (#), space ( ), and underscore (_).
4. For most features, you associate an action or a profile. For example, you may be required to select an action, or, in the case of an NetScaler Gateway or application firewall policy, you select a profile to associate with the policy. A profile is a set of configuration options that operate as a set of actions that are applied when the data being analyzed matches the policy rule.
5. Create an expression that describes the type of data that you want this policy to match.

    Depending on the type of policy you want to create, you can choose a predefined expression, or you can create a new expression.

    Named expressions are predefined expressions that you can reference by name in a policy rule.

6. Click Create to create your new policy.
7. Click Close to return to the Policies screen for the type of policy you were creating.

Configure a classic expression

September 21, 2020

Classic expressions consist of the following expression elements, listed in hierarchical order:

- **Flow Type.** Specifies whether the connection is incoming or outgoing. The flow type is REQ for incoming connections and RES for outgoing connections.
- **Protocol.** Specifies the protocol, the choices for which are HTTP, SSL, TCP, and IP.
- **Qualifier.** The protocol attribute, which depends on the selected protocol.
- **Operator.** The type of test you want to perform on the connection data. Your choice of operator depends upon the connection information you are testing. If the connection information you are testing is text, you use text operators. If it is a number, you use standard numeric operators.
- **Value.** The string or number against which the connection data element—defined by the flow type, protocol, and qualifier—is tested. The value can be either a literal or an expression. The literal or expression must match the data type of the connection data element.

In a policy, classic expressions can be combined to create more complex expressions using Boolean and comparative operators.

Expression elements are parsed from left to right. The leftmost element is either REQ or RES and designates a request or a response, respectively. Successive terms define a specific connection type and a specific attribute for that connection type. Each term is separated from any preceding or following term by a period. Arguments appear in parentheses and follow the expression element to which they are passed.

The following classic expression fragment returns the client source IP for an incoming connection.

```
REQ.IP.SOURCEIP
```

The example identifies an IP address in a request. The expression element SOURCEIP designates the source IP address. This expression fragment may not be useful by itself. You can use an additional expression element, an operator, to determine whether the returned value meets specific criteria. The following expression tests whether the client IP is in the subnet 200.0.0.0/8 and returns a Boolean TRUE or FALSE:

```
REQ.IP.SOURCEIP == 200.0.0.0 -netmask 255.0.0.0
```

Create a classic policy expression by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  set appfw policy \<name\> -rule \<expression\> -action \<action\>

-  show appfw policy \<name\>
<!--NeedCopy-->
```

Example

```
> set appfw policy GenericApplicationSSL_ 'HTTP.REQ.METHOD.EQ("get")' APPFW_DROP
 Done
> show appfw policy GenericApplicationSSL_
        Name: GenericApplicationSSL_    Rule: HTTP.REQ.METHOD.EQ("get")
        Profile: APPFW_DROP     Hits: 0
        Undef Hits: 0
        Policy is bound to following entities
        1) REQ VSERVER app_u_GenericApplicationSSLPortalPages PRIORITY : 100
 Done
<!--NeedCopy-->
```

Add an expression for a classic policy by using the GUI

This procedure documents the Add Expression dialog box. Depending on the feature for which you are configuring a policy, the route by which you arrive at this dialog box may be different.

1. Perform steps 1-4 in "To create a policy with classic expressions by using the GUI."
2. In the Add Expression dialog box, in Expression Type, click the type of expression you want to create.
3. Under Flow Type, click the down arrow and choose a flow type.

   The flow type is typically REQ or RES. The REQ option specifies that the policy applies to all incoming connections or requests. The RES option applies the policy to all outgoing connections or responses.

   For Application Firewall policies, you should leave the expression type set to General Expression, and the flow type set to REQ. The Application Firewall treats each request and response as a single paired entity, so all Application Firewall policies begin with REQ.

4. Under Protocol, click the down arrow and choose the protocol you want for your policy expression. Your choices are:
    - HTTP. Evaluates HTTP requests that are sent to a Web server. For classic expressions, HTTP includes HTTPS requests.
    - SSL. Evaluates SSL data associated with the current connection.
    - TCP. Evaluates the TCP data associated with the current connection.
    - IP. Evaluates the IP addresses associated with the current connection.
5. Under Qualifier, click the down arrow and choose a qualifier for your policy.

    The qualifier defines the type of data to be evaluated. The list of qualifiers that appears depends on which protocol you selected in step 4.

    The following choices appear for the HTTP protocol:

    - METHOD. Filters HTTP requests that use a particular HTTP method.
    - URL. Filters HTTP requests for a specific Web page.
    - URLQUERY. Filters HTTP requests that contain a particular query string.
    - VERSION. Filters HTTP requests on the basis of the specified HTTP protocol version.
    - HEADER. Filters on the basis of a particular HTTP header.
    - URLLEN. Filters on the basis of the length of the URL.
    - URLQUERY. Filters on the basis of the query portion of the URL.
    - URLQUERYLEN. Filters on the basis of the length of the query portion of the URL only.
6. Under Operator, click the down arrow and choose the operator for your policy expression. Some common operators are:

Operator	Description
==	Matches the specified value exactly or is exactly equal to the specified value.
!=	Does not match the specified value.
>	Is greater than the specified value.
<	Is less than the specified value.
>=	Is greater than or equal to the specified value.
<=	Is less than or equal to the specified value.
CONTAINS	Contains the specified value.
CONTENTS	Returns the contents of the designated header, URL, or URL query.
EXISTS	The specified header or query exists.

Operator	Description
NOTCONTAINS	Does not contain the specified value.
NOTEXISTS	The specified header or query does not exist.

7. If a Value text box appears, type a string or numeric value, as appropriate. For example, chose REQ as the Flow Type, HTTP as the Protocol, and HEADER as the qualifier, and then type the value of the header string in the Value field and the header type for which you want to match the string in the Header Name text box.
8. Click OK.
9. To create a compound expression, click Add. Note that the type of compounding that is done depends on the following choices in the Create Policy dialog box:
    - **Match Any Expression.** The expressions are in a logical OR relationship.
    - **Match All Expressions.** The expressions are in a logical AND relationship.
    - **Tabular Expressions.** Click the AND, OR, and parentheses buttons to control evaluation.
    - **Advanced Free-Form.** Enter the expressions components directly into the Expression field, and click the AND, OR, and parentheses buttons to control evaluation.

Bind a classic policy

September 21, 2020

Depending on the policy type, you can bind a classic policy either globally or to a virtual server. Policy bind points are described in the table, "Policy Type and Bind Points for Policies in Features That Use Classic Policies."

Note: You can bind a classic policy to multiple bind points.

Bind a classic policy globally by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  bind cmp global <policyName> [-priority <positive_integer>]

-  show cmp global
<!--NeedCopy-->
```

Example

```
> bind cmp global cmp-pol-compress -priority 2
 Done
> show cmp global
1)      Policy Name: cmp-pol-compress   Priority: 2
2)      Policy Name: ns_nocmp_xml_ie    Priority: 8700
3)      Policy Name: ns_nocmp_mozilla_47        Priority: 8800
4)      Policy Name: ns_cmp_mscss       Priority: 8900
5)      Policy Name: ns_cmp_msapp       Priority: 9000
6)      Policy Name: ns_cmp_content_type        Priority: 10000
 Done
>
<!--NeedCopy-->
```

Bind a classic policy to a virtual server by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  bind lb vserver <name> [<targetVserver>] [-policyName <string>] [-priority <positive_integer>]

-  show lb vserver<name>
<!--NeedCopy-->
```

Example

```
> bind lb vserver lbtemp -policyName cmp-pol-compress -priority 1
 Done
> show lb vserver lbtemp
        lbtemp (10.102.29.101:80) - HTTP        Type: ADDRESS
        State: UP
        Last state change was at Tue Oct 27 06:40:38 2009 (+557 ms)
        Time since last state change: 0 days, 02:00:40.330
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state changed to UP
        Group: vserver-grp
        Mode: IP
        Persistence: COOKIEINSERT (version 0)   Persistence Backup: SOURCEIP   Persistence Mask: 255.255.255.255
        Persistence Timeout: 2 min      Backup Persistence Timeout: 2 min
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
1) http-one (10.102.29.252: 80) - HTTP State: UP        Weight: 1
        Persistence Cookie Value : NSC_wtfswfs-hsq=ffffffff096e03ed45525d5f4f58455e445a4a423660
1)      Policy : cmp-pol-compress Priority:1
 Done
>
<!--NeedCopy-->
```

Bind a classic policy globally by using the GUI

Note: This procedure documents the Global Bindings dialog box. Depending on the feature for which you want to globally bind a policy, the route by which you arrive at this dialog box may be different.

1. In the navigation pane, expand the feature for which you want to globally bind a classic policy, and then locate the policy that you want to bind globally.

   Note: You cannot globally bind policies for Content Switching, Cache Redirection, SureConnect, Priority Queuing, or
   NetScaler Gateway Authorization.

2. In the details pane, click Global Bindings.
3. In the Bind/Unbind <feature name> Policy(s) to Global dialog box, click Insert Policy.
4. In the Policy Name column, click the name of an existing policy that you want to globally bind, or click New Policy to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, type the priority value.

   The lower the number, the sooner this policy is applied relative to other policies. For example, a policy assigned a priority of 10 is applied before a policy with a priority of 100. You can use the same priority for different policies. All features that use classic policies implement only the first policy that a connection matches, so policy priority is important for getting the results you intend.

As a best practice, leave room to add policies by setting priorities with intervals of 50 (or 100) between each policy.

6. Click OK.

Bind a classic policy to a virtual server by using the GUI

1. In the navigation pane, expand the feature that contains the virtual server to which you want to bind a classic policy (for example, if you want to bind a classic policy to a content switching virtual server, expand Traffic Management > Content Switching), and then click Virtual Servers.
2. In the details pane, select the virtual server, and then click Open.
3. In the Configure <Feature> Virtual Server dialog box, on the Policies tab, click the feature icon for the type policy that you want, and then click Insert Policy.
4. In the Policy Name column, click the name of an existing policy that you want to bind to a virtual server, or click A to open the Create <feature name> Policy dialog box.
5. After you have selected the policy or created a new policy, in the Priority column, set the priority.

   If you are binding a policy to a content switching virtual server, in the Target column, select a load balancing virtual server to which traffic that matches the policy should be sent.
6. Click OK.

View classic policies

September 21, 2020

You can view classic policies by using either the GUI or the command line. You can view details such as the policy's name, expression, and bindings.

View a classic policy and its binding information by using the CLI

At the command prompt, type the following commands to view a classic policy and its binding information:

```
show <featureName> policy [policyName]
```

Example

```
> show appfw policy GenericApplicationSSL_
        Name: GenericApplicationSSL_    Rule: ns_only_get_adv
        Profile: GenericApplicationSSL_Prof1    Hits: 0
        Undef Hits: 0
        Policy is bound to following entities
        1) REQ VSERVER app_u_GenericApplicationSSLPortalPages
           PRIORITY : 100
 Done
<!--NeedCopy-->
```

Note: If you omit the policy name, all policies are listed without the binding details.

View classic policies and policy bindings by using the GUI

1. In the navigation pane, expand the feature whose policies you want to view, (for example, if you want to view application firewall policies, expand Application Firewall), and then click Policies.
2. In the details pane, do one or more of the following:
    - To view details for a specific policy, click the policy. Details appear in the Details area of the configuration pane.
    - To view bindings for a specific policy, click the policy, and then click Show Bindings.
    - To view global bindings, click the policy, and then click Global Bindings. Note that you cannot bind a Content Switching, Cache Redirection, SureConnect, Priority Queuing, or NetScaler Gateway Authorization policy globally.

Create named classic expressions

September 21, 2020

A named classic expression is a classic expression that can be referenced through an assigned name. Often, you need to configure classic expressions that are large or complex and form a part of a larger compound expression. You might also configure classic expressions that you need to use frequently and in multiple compound expressions or classic policies. In these scenarios, you can create the classic expression you want, save it with a name of your choice, and then reference the expression from compound expressions or policies through its name. This saves configuration time and improves the readability of complex compound expressions. Additionally, any modifications to a named classic expression need to be made only once.

Some named expressions are built-in, and a subset of these are read-only. Built-in named expressions are divided into four categories: General, Anti-Virus, Personal Firewall, and Internet Security. General named expressions have a wide variety of uses. For example, from the General category, you can use the expressions ns_true and ns_false to specify a value of TRUE or FALSE, respectively, to be returned for all traffic. You can also identify data of a particular type (for example, HTM, DOC, or GIF files), determine whether caching headers are present, or determine whether the round trip time for packets between a client and the NetScaler is high (over 80 milliseconds).

Anti-Virus, Personal Firewall, and Internet Security named expressions test clients for the presence of a particular program and version and are used primarily in NetScaler Gateway policies.

Note: You cannot modify or delete built-in named expressions.

Create a named classic expression by using the CLI

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  add expression <name> <value> [-comment <string>] [-clientSecurityMessage <string>]
-  show expression [<name> | -type CLASSIC
<!--NeedCopy-->
```

Example

```
> add expression classic_ne "REQ.HTTP.URL CONTAINS www.example1.com" -comment "Checking the URL for www.example1.com"
 Done
> show expression classic_ne
1)       Name: classic_ne  Expr: REQ.HTTP.URL CONTAINS www.example1.com Hits: 0 Type : CLASSIC
         Comment: "Checking the URL for www.example1.com"
 Done
>
<!--NeedCopy-->
```

Create a named classic expression by using the GUI

1. In the navigation pane, expand AppExpert, expand Expressions, and then click Classic Expressions.
2. In the details pane, click Add.

   Note: Some of the built-in expressions in the Expressions list are read-only.
3. In the Create Policy Expression dialog box, specify values for the following parameters:

- Expression Name*—name
- Client Security Message—clientSecurityMessage
- Comments—comment

*A required parameter

4. To create the expression, do one of the following:
    - You can choose inputs to this expression from the Named Expressions drop-down list.
    - You can create a new expression, as described in Add an expression for a classic policy by using the GUI.
5. When you are done, click Close. Verify that your new expression was created by scrolling to the bottom of the Classic Expressions list to view it.

Expressions reference-advanced policy expressions

September 21, 2020

> **Warning**
>
> Q and S prefixes are deprecated from NetScaler 12.0 build 56.20 onwards and are no longer supported in Advanced policy expressions.

The following table is a listing of default syntax expression prefixes, with cross-references to descriptions of these prefixes and the operators that you can specify for them. Note that some prefixes can work with multiple types of operators. For example, a cookie can be parsed by using operators for text or operators for HTTP headers.

You can use any element in the following tables as a complete expression on its own, or you can use various operators to combine these expression elements with others to form more complex expressions.

Note: The Description column in the following table contains cross-references to additional information about prefix usage and applicable operators for the prefix.

Click here to view the table.

Expressions reference-classic expressions

September 21, 2020

> **Warning**
>
> Classic policy expressions are no longer supported from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use Advanced policies. For more information, see Advanced Policies

The subtopics listed in the table of contents on the left side of your screen contain tables listing the NetScaler classic expressions.

In the table of operators, the result type of each operator is shown at the beginning of the description. In the other tables, the level of each expression is shown at the beginning of the description. For named expressions, each expression is shown as a whole.

Operators

Expression Element	Definition
==	Boolean. Returns TRUE if the current expression equals the argument. For text operations, the items being compared must exactly match one another. For numeric operations, the items must evaluate to the same number.
!=	Boolean. Returns TRUE if the current expression does not equal the argument. For text operations, the items being compared must not exactly match one another. For numeric operations, the items must not evaluate to the same number.
CONTAINS	Boolean. Returns TRUE if the current expression contains the string that is designated in the argument.
NOTCONTAINS	Boolean. Returns TRUE if the current expression does not contain the string that is designated in the argument.
CONTENTS	Text. Returns the contents of the current expression.
EXISTS	Boolean. Returns TRUE if the item designated by the current expression exists.
NOTEXISTS	Boolean.

Expression Element	Definition
Returns TRUE if the item designated by the current expression does not exist.	
>	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than the argument.
<	Boolean. Returns TRUE if the current expression evaluates to a number that is less than the argument.
>=	Boolean. Returns TRUE if the current expression evaluates to a number that is greater than or equal to the argument.
<=	Boolean. Returns TRUE if the current expression evaluates to a number that is less than or equal to the argument.

General expressions

Expression Element	Definition
REQ	Flow Type. Operates on incoming (or request) packets.
REQ.HTTP	Protocol. Operates on HTTP requests.
REQ.HTTP.METHOD	Qualifier. Designates the HTTP method.
REQ.HTTP.URL	Qualifier. Designates the URL.
REQ.HTTP.URLTOKENS	Qualifier. Designates the URL token.
REQ.HTTP.VERSION	Qualifier. Designates the HTTP version.
REQ.HTTP.HEADER	Qualifier. Designates the HTTP header.
REQ.HTTP.URLLEN	Qualifier. Designates the number of characters in the URL.
REQ.HTTP.URLQUERY	Qualifier. Designates the query portion of the URL.
REQ.HTTP.URLQUERYLEN	Qualifier. Designates the length of the query portion of the URL.
REQ.SSL	Protocol. Operates on SSL requests.

Expression Element	Definition
REQ.SSL.CLIENT.CERT	Qualifier. Designates the entire client certificate.
REQ.SSL.CLIENT.CERT.SUBJECT	Qualifier. Designates the client certificate subject.
REQ.SSL.CLIENT.CERT.ISSUER	Qualifier. Designates the issuer of the client certificate.
REQ.SSL.CLIENT.CERT.SIGALGO	Qualifier. Designates the validation algorithm used by the client certificate.
REQ.SSL.CLIENT.CERT.VERSION	Qualifier. Designates the client certificate version.
REQ.SSL.CLIENT.CERT.VALIDFROM	Qualifier. Designates the date before which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.VALIDTO	Qualifier. Designates the date after which the client certificate is not valid.
REQ.SSL.CLIENT.CERT.SERIALNUMBER	Qualifier. Designates the serial number of the client certificate.
REQ.SSL.CLIENT.CIPHER.TYPE	Qualifier. Designates the encryption protocol used by the client.
REQ.SSL.CLIENT.CIPHER.BITS	Qualifier. Designates the number of bits used by the client's SSL key.
REQ.SSL.CLIENT.SSL.VERSION	Qualifier. Designates the SSL version that the client is using.
REQ.TCP	Protocol. Operates on incoming TCP packets.
REQ.TCP.SOURCEPORT	Qualifier. Designates the source port of the incoming packet.
REQ.TCP.DESTPORT	Qualifier. Designates the destination port of the incoming packet.
REQ.IP	Protocol. Operates on incoming IP packets.
REQ.IP.SOURCEIP	Qualifier. Designates the source IP of the incoming packet.
REQ.IP.DESTIP	Qualifier.Designates the destination IP of the incoming packet.
RES	Flow Type. Operates on outgoing (or response) packets.

Expression Element	Definition
RES.HTTP	Protocol. Operates on HTTP responses.
RES.HTTP.VERSION	Qualifier. Designates the HTTP version.
RES.HTTP.HEADER	Qualifier. Designates the HTTP header.
RES.HTTP.STATUSCODE	Qualifier. Designates the status code of the HTTP response.
RES.TCP	Protocol. Operates on incoming TCP packets.
RES.TCP.SOURCEPORT	Qualifier. Designates the source port of the outgoing packet.
RES.TCP.DESTPORT	Qualifier. Designates the destination port of the outgoing packet.
RES.IP	Protocol. Operates on outgoing IP packets.
RES.IP.SOURCEIP	Qualifier. Designates the source IP of the outgoing packet. This can be in IPv4 or IPv6 format. For example: add expr exp3 “sourceip == 10.102.32.123 –netmask 255.255.255.0 && destip == 2001::23/120”.
RES.IP.DESTIP	Qualifier. Designates the destination IP of the outgoing packet.

Client security expressions

The expressions to configure client settings on the Access Gateway with the following software:

- Antivirus
- Personal firewall
- Antispam
- Internet Security

For example usage, see http://support.citrix.com/article/CTX112599.

Actual Expression	Definition
CLIENT.APPLICATION.AV(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-virus program and version.
CLIENT.APPLICATION.AV(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-virus program and version.

Actual Expression	Definition
CLIENT.APPLICATION.PF(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated personal firewall program and version.
CLIENT.APPLICATION.PF(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated personal firewall program and version.
CLIENT.APPLICATION.IS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated internet security program and version.
CLIENT.APPLICATION.IS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated internet security program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION == <VERSION>)	Checks whether the client is running the designated anti-spam program and version.
CLIENT.APPLICATION.AS(<NAME>.VERSION != <VERSION>)	Checks whether the client is not running the designated anti-spam program and version.

Network-based expressions

Expression	Definition
REQ	Flow Type. Operates on incoming, or request, packets.
REQ.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
REQ.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
REQ.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
REQ.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.

Expression	Definition
REQ.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
REQ.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
REQ.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.
RES	Flow Type. Operates on outgoing (or response) packets.
RES.VLANID	Qualifier. Operates on the virtual LAN (VLAN) ID.
RES.INTERFACE.ID	Qualifier. Operates on the ID of the designated NetScaler interface.
RES.INTERFACE.RXTHROUGHPUT	Qualifier. Operates on the raw received packet throughput of the designated NetScaler interface.
RES.INTERFACE.TXTHROUGHPUT	Qualifier. Operates on the raw transmitted packet throughput of the designated NetScaler interface.
RES.INTERFACE.RXTXTHROUGHPUT	Qualifier. Operates on the raw received and transmitted packet throughput of the designated NetScaler interface.
RES.ETHER.SOURCEMAC	Qualifier. Operates on the source MAC address.
RES.ETHER.DESTMAC	Qualifier. Operates on the destination MAC address.

Date/time expressions

Expression	Definition
TIME	Qualifier. Operates on the date and time of day, GMT.
DATE	Qualifier. Operates on the date, GMT.
DAYOFWEEK	Operates on the specified day in the week, GMT.

File system expressions

You can specify file system expressions in authorization policies for users and groups who access file sharing through the NetScaler Gateway file transfer utility (the VPN portal). These expressions work with the NetScaler Gateway file transfer authorization feature to control user access to file servers, folders, and files. For example, you can use these expressions in authorization policies to control access based on file type and size.

For more information, refer to the File Name Expression pdf.

Note: File system expressions do not support regular expressions.

Built-in named expressions (General)

Expression	Definition
ns_all_apps_ncomp	Tests for connections with destination ports between 0 and 65535. In other words, tests for all applications.
ns_cachecontrol_nocache	Tests for connections with an HTTP Cache-Control header that contains the value “no-cache”.
ns_cachecontrol_nostore	Tests for connections with an HTTP Cache-Control header that contains the value “no-store”.
ns_cmpclient	Tests the client to determine if it accepts compressed content.
ns_content_type	Tests for connections with an HTTP Content-Type header that contains “text”.
ns_css	Tests for connections with an HTTP Content-Type header that contains “text/css”.
ns_ext_asp	Tests for HTTP connections to any URL that contains the string .asp—in other words, any connection to an active server page (ASP).
ns_ext_cfm	Tests for HTTP connections to any URL that contains the string .cfm
ns_ext_cgi	Tests for HTTP connections to any URL that contains the string .cgi—in other words, any connection to a common gateway interface (CGI) script.

Expression	Definition
ns_ext_ex	Tests for HTTP connections to any URL that contains the string .ex
ns_ext_exe	Tests for HTTP connections to any URL that contains the string .exe—in other words, any connection to a executable file.
ns_ext_htx	Tests for HTTP connections to any URL that contains the string .htx
ns_ext_not_gif	Tests for HTTP connections to any URL that does not contain the string .png—in other words, any connection to a URL that is not a GIF image.
ns_ext_not_jpeg	Tests for HTTP connections to any URL that does not contain the string .jpeg—in other words, any connection to a URL that is not a JPEG image.
ns_ext_shtml	Tests for HTTP connections to any URL that contains the string .shtml—in other words, any connection to a server-parsed HTML page.
ns_false	Always returns a value of FALSE.
ns_farclient	Client is in a different geographical region from the NetScaler, as determined by the geographical region in the client's IP address. The following regions are predefined: 192.0.0.0 – 193.255.255.255: Multi-regional, 194.0.0.0 – 195.255.255.255: European Union, 196.0.0.0 – 197.255.255.255: Other1, 198.0.0.0 – , 199.255.255.255: North America, 200.0.0.0 – 201.255.255.255: Central and South America, 202.0.0.0 – 203.255.255.255: Pacific Rim, 204.0.0.0 – 205.255.255.255: Other2, and 206.0.0.0 – 207.255.255.255: Other3
ns_header_cookie	Tests for HTTP connections that contain a Cookie header.
ns_header_pragma	Tests for HTTP connections that contain a Pragma: no-cache header.

Expression	Definition
ns_mozilla_47	Tests for HTTP connections whose User-Agent header contains the string Mozilla/4.7—in other words, any connection from a client using the Mozilla 4.7 Web browser.
ns_msexcel	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msexcel—in other words, any connection transmitting a Microsoft Excel spreadsheet.
ns_msie	Tests for HTTP connections whose User-Agent header contains the string MSIE—in other words, any connection from a client using any version of the Internet Explorer Web browser.
ns_msppt	Tests for HTTP connections whose Content-Type header contains the string application/vnd.ms-powerpoint—in other words, any connection transmitting a Microsoft PowerPoint file.
ns_msword	Tests for HTTP connections whose Content-Type header contains the string application/vnd.msword—in other words, any connection transmitting a Microsoft Word file.
ns_non_get	Tests for HTTP connections that use any HTTP method except for GET.
ns_slowclient	Returns TRUE if the average round trip time between the client and the NetScaler is more than 80 milliseconds.
ns_true	Returns TRUE for all traffic.
ns_url_path_bin	Tests the URL path to see if it points to the /bin/ directory.
ns_url_path_cgibin	Tests the URL path to see if it points to the CGI-BIN directory.
ns_url_path_exec	Tests the URL path to see if it points to the /exec/directory.
ns_url_tokens	Tests for the presence of URL tokens.

Expression	Definition
ns_xmldata	Tests for the presence of XML data.

Built-in named expressions (Anti-Virus)

Expression	Definition
McAfee Virus Scan 11	Tests to determine whether the client is running the latest version of McAfee VirusScan.
McAfee Antivirus	Tests to determine whether the client is running any version of McAfee Antivirus.
Symantec AntiVirus 10 (with Updated Definition File)	Tests to determine whether the client is running the most current version of Symantec AntiVirus.
Symantec AntiVirus 6.0	Tests to determine whether the client is running Symantec AntiVirus 6.0.
Symantec AntiVirus 7.5	Tests to determine whether the client is running Symantec AntiVirus 7.5.
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
TrendMicro AntiVirus 11.25	Tests to determine whether the client is running Trend Microsystems' AntiVirus, version 11.25.
Sophos Antivirus 4	Tests to determine whether the client is running Sophos Antivirus, version 4.
Sophos Antivirus 5	Tests to determine whether the client is running Sophos Antivirus, version 5.
Sophos Antivirus 6	Tests to determine whether the client is running Sophos Antivirus, version 6.

Built-in named expressions (Personal Firewall)

Expression	Definition
TrendMicro OfficeScan 7.3	Tests to determine whether the client is running Trend Microsystems' OfficeScan, version 7.3.
Sygate Personal Firewall 5.6	Tests to determine whether the client is running the Sygate Personal Firewall, version 5.6.
ZoneAlarm Personal Firewall 6.5	Tests to determine whether the client is running the ZoneAlarm Personal Firewall, version 6.5.

Built-in named expressions (Client Security)

Expression	Definition
Norton Internet Security	Tests to determine whether the client is running any version of Norton Internet Security.

Summary examples of default syntax expressions and policies

September 21, 2020

The following table provides examples of default syntax expressions that you can use as the basis for your own default syntax expressions.

Table 1. Examples of Default Syntax Expressions

Expression Type	Sample Expressions
Look at the method used in the HTTP request.	`http.req.method.eq(post)http.req.method.eq(get)`

Expression Type	Sample Expressions

Check the Cache-Control or Pragma header value in an HTTP request (req) or response (res).	`http.req.header("Cache-Control").contains("no-store")`
	`http.req.header("Cache-Control").contains("no-cache")`
	`http.req.header("Pragma").contains("no-cache")`
	`http.res.header("Cache-Control").contains("private")`
	`http.res.header("Cache-Control").contains("public")`
	`http.res.header("Cache-Control").contains("must-revalidate")http.res.header("Cache-Control").contains ("proxy-revalidate")`
	`http.res.header("Cache-Control").contains("max-age")`

Check for the presence of a header in a request (req) or response (res).	`http.req.header("myHeader").exists` `http.res.header("myHeader").exists`
Look for a particular file type in an HTTP request based on the file extension.	`http.req.url.contains(".html")http.req.url.contains(".cgi")http.req.url.contains(".asp")http.req.url.contains(".exe")http.req.url.contains(".cfm")http.req.url.contains(".ex")http.req.url.contains(".shtml")http.req.url.contains(".htx")http.req.url.contains("/cgi-bin/")http.req.url.contains("/exec/")http.req.url.contains("/bin/")`
Look for anything that is other than a particular file type in an HTTP request.	`http.req.url.contains(".png").not;` `http.req.url.contains(".jpeg").not`

Expression Type	Sample Expressions
Check the type of file that is being sent in an HTTP response based on the Content-Type header.	`http.res.header("Content-Type").contains("text")http.res.header("Content-Type").contains "application/msword")http.res.header("Content-Type").contains("vnd.ms-excel")http.res.header("Content-Type").contains("application/vnd.ms-powerpoint"); http.res.header("Content-Type").contains("text/css"); http.res.header("Content-Type").contains("text/xml"); http.res.header("Content-Type").contains("image/")`
Check whether this response contains an expiration header.	‘http.res.header(“Expires”).exists
Check for a Set-Cookie header in a response.	http.res.header(“Set-Cookie”).exists
Check the agent that sent the response.	http.res.header(“User-Agent”).contains(“Mozilla/4.7”) http.res.header(“User-Agent”).contains(“MSIE”)‘
Check if the first 1024 bytes of the body of a request starts with the string “some text”.	`http.req.body(1024).contains("some text")`

The following table shows examples of policy configurations and bindings for commonly used functions.

Table 2. Examples of Default Syntax Expressions and Policies

Purpose	Example
Use the rewrite feature to replace occurrences of `http://with https://` in the body of an HTTP response.	`add rewrite action httpRewriteAction replace_all http.res.body(50000)"\"https://\""-pattern http://add rewrite policy demo_rep34312 "http.res.body(50000).contains(\"http://\")" httpRewriteAction`
Replace all occurrences of “abcd” with “1234” in the first 1000 bytes of the HTTP body.	`add rewrite action abcdTo1234Action replace_all "http.req.body(1000)"" \"1234\""-pattern abcd add rewrite policy abcdTo1234Policy "http.req.body(1000).contains(\"abcd\")" abcdTo1234Action bind rewrite global abcdTo1234Policy 100 END -type REQ_OVERRIDE`
Downgrade the HTTP version to 1.0 to prevent the server from chunking HTTP responses.	`add rewrite action downgradeTo1.0 Action replace http.req.version.minor "\"0\""add rewrite policy downgradeTo1.0Policy "http.req.version.minor.eq(1)"downgradeTo1.0 Action bind lb vserver myLBVserver -policyName downgradeTo1.0Policy -priority 100 -gotoPriorityExpression NEXT -type REQUEST`
Remove references to the HTTP or HTTPS protocol in all responses, so that if the user’s connection is HTTP, the link is opened by using HTTP, and if the user’s connection is HTTPS, the link is opened by using HTTPS.	`add rewrite action remove_http_https replace_all "http.res.body(1000000).set_text_mode(ignorecase)""\"//\""-pattern "re~https?://|HTTPS?://~"add rewrite policy remove_http_https true remove_http_https bind lb vserver test_vsvr -policyName remove_http_https -priority 20 -gotoPriorityExpression NEXT -type RESPONSE`

Purpose	Example
Rewrite instances of http: to https: in all URLs.	`add responder action httpToHttpsAction redirect "\"https://\" + http.req.hostname + http.req.url"-bypassSafetyCheck YES add responder policy httpToHttpsPolicy "!CLIENT.SSL.IS_SSL" httpToHttpsAction bind responder global httpToHttpsPolicy 1 END -type OVERRIDE`
Modify a URL to redirect from URL A to URL B. In this example, "file5.html" is appended to the path.	`add responder action appendFile5Action redirect \"http://\" + http.req.hostname + http.req.url + \"/file5.html\""-bypassSafetyCheck YES add responder policy appendFile5Policy "http.req.url.eq(\"/testsite\")" appendFile5Action bind responder global appendFile5Policy 1 END -type OVERRIDE`
Redirect an external URL to an internal URL.	`add rewrite action act_external_to_internal REPLACE 'http.req.hostname.server''"www.my.host.com"'add rewrite policy pol_external_to_internal 'http.req.hostname.server.eq("www.external.host.com")'act_external_to_internal bind rewrite global pol_external_to_internal 100 END -type REQ_OVERRIDE`

Purpose	Example
Redirect requests to www.example.com that have a query string to www.Webn.example.com. The value n is derived from a server parameter in the query string, for example, server=5.	`add rewrite action act_redirect_query REPLACE q##http.req.header("Host").before_str(".example.com")''"Web"+ http.req.url.query.value("server")## add rewrite policy pol_redirect_query q##http.req.header("Host").eq("www.example.com")&& http.req.url.contains("?")'act_redirect_query##`
Limit the number of requests per second from a URL.	`add ns limitSelector ip_limit_selector http.req.url "client.ip.src"add ns limitIdentifier ip_limit_identifier -threshold 4 -timeSlice 3600 -mode request_rate -limitType smooth -selectorName ip_limit_selector add responder action my_Web_site_redirect_action redirect "\"http://www.mycompany.com/\""add responder policy ip_limit_responder_policy "http.req.url.contains(\"myasp.asp\")&& sys.check_limit (\"ip_limit_identifier\")"my_Web_site_redirect_action bind responder global ip_limit_responder_policy 100 END -type default`
Check the client IP address but pass the request without modifying the request.	`add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER ("x-forwarded-for").EXISTS HTTP.REQ.HEADER ("client-ip").EXISTS'NOREWRITE bind rewrite global check_client_ip_policy 100 END`

Purpose	Example
Remove old headers from a request and insert an NS-Client header.	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END`

Purpose	Example
Remove old headers from a request, insert an NS-Client header, and then modify the "insert header" action so that the value of the inserted header contains the client IP values from the old headers and the NetScaler appliance's connection IP address. Note that this example repeats the previous example, with the exception of the final set rewrite action.	`add rewrite action del_x_forwarded_for delete_http_header x-forwarded-for add rewrite action del_client_ip delete_http_header client-ip add rewrite policy check_x_forwarded_for_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS'del_x_forwarded_for add rewrite policy check_client_ip_policy 'HTTP.REQ.HEADER("client-ip").EXISTS' del_client_ip add rewrite action insert_ns_client_header insert_http_header NS-Client 'CLIENT.IP.SRC'add rewrite policy insert_ns_client_policy 'HTTP.REQ.HEADER("x-forwarded-for").EXISTS HTTP.REQ.HEADER("client-ip").EXISTS 'insert_ns_client_header bind rewrite global check_x_forwarded_for_policy 100 200 bind rewrite global check_client_ip_policy 200 300 bind rewrite global insert_ns_client_policy 300 END set rewrite action insert_ns_client_header -stringBuilderExpr 'HTTP.REQ.HEADER("x-forwarded-for").VALUE(0)+ " " + HTTP.REQ.HEADER("client-ip").VALUE(0)+ " " + CLIENT.IP.SRC'-bypassSafetyCheck YES`

Tutorial examples of default syntax policies for rewrite

September 21, 2020

With the rewrite feature, you can modify any part of an HTTP header, and, for responses, you can modify the HTTP body. You can use this feature to accomplish a number of useful tasks, such as removing unnecessary HTTP headers, masking internal URLs, redirecting Web pages, and redirecting queries or keywords.

In the following examples, you first create a rewrite action and a rewrite policy. Then you bind the policy globally.

This document includes the following details:

- Redirecting an External URL to an Internal URL
- Redirecting a Query
- Rewriting HTTP to HTTPS
- Removing Unwanted Headers
- Reducing Web Server Redirects
- Masking the Server Header

For more information about the commands and syntax descriptions, see Rewrite Command Reference page.

Redirecting an external URL to an internal URL

This example describes how to create a rewrite action and rewrite policy that redirects an external URL to an internal URL. You create an action, called act_external_to_internal, that performs the rewrite. Then you create a policy called pol_external_to_internal.

To redirect an external URL to an internal URL by using the command line interface

- To create the rewrite action, at the command prompt, type:

```
add rewrite action act_external_to_internal REPLACE "http.req.hostname.
server""host_name_of_internal_Web_server"
```

- To create the rewrite policy, at the Citrix ADC command prompt, type:

```
add rewrite policy pol_external_to_internal "http.req.hostname.server.eq("
host_name_of_external_Web_server")act_external_to_internal"
```

- Bind the policy globally.

To redirect an external URL to an internal URL by using the GUI

1. Navigate to **AppExpert** > **Rewrite** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Rewrite Action** dialog box, enter the name **act_external_to_internal**.
4. To replace the HTTP server host name with the internal server name, choose Replace from the Type list box.
5. In the **Header Name** field, type Host.
6. In the **String expression for replacement text** field, type the internal host name of your Web server.
7. Click **Create** and then click **Close**.
8. In the navigation pane, click **Policies**.
9. In the details pane, click **Add**.
10. In the **Name** field, type **pol_external_to_internal**. This policy will detect connections to the Web server.
11. In the **Action** drop-down menu, choose the **action act_external_to_internal**.
12. In the **Expression editor**, construct the following expression:

```
HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")
<!--NeedCopy-->
```

1. Bind your new policy globally.

Redirecting a query

This example describes how to create a rewrite action and rewrite policy that redirects a query to the proper URL. The example assumes that the request contains a Host header set to **www.example.com** and a GET method with the **string /query.cgi?server=5**. The redirect extracts the domain name from the host header and the number from the query string, and redirects the user's query to the server **Web5.example.com**, where the rest of the user's query is processed.

> **Note:**
>
> Although the following commands appears on multiple lines, you should enter them on a single line without line breaks.

To redirect a query to the appropriate URL using the command line

- To create a rewrite action named act_redirect_query that replaces the HTTP server host name with the internal server name, type:

```
add rewrite action act_redirect_query REPLACE q##http.req.header("Host").
before_str(".example.com")''"Web"+ http.req.url.query.value("server")##
```

- To create a rewrite policy named pol_redirect_query, type the following commands at the NetScaler command prompt.. This policy detects connections, to the Web server, that contain a query string. Do not apply this policy to connections that do not contain a query string:

```
add rewrite policy pol_redirect_query q##http.req.header("Host").eq("www.
example.com")&& http.req.url.contains("?")'act_redirect_query##
```

- Bind your new policy globally.

Because this rewrite policy is highly specific and should be run before any other rewrite policies, it is advisable to assign it a high priority. If you assign it a priority of 1, it will be evaluated first.

Rewriting HTTP to HTTPS

This example describes how to rewrite Web server responses to find all URLs that begin with the string "http" and replace that string with "https." You can use this to avoid having to update Web pages after moving a server from HTTP to HTTPS.

To redirect HTTP URLs to HTTPS by using the command line interface

- To create a rewrite action named act_replace_http_with_https that replaces all instances of the string "http" with the string "https," enter the following command:

```
add rewrite action act_replace_http_with_https replace_all 'http.res.body
(100)''"https"'-pattern http
```

- To create a rewrite policy named pol_replace_http_with_https that detects connections to the Web server, enter the following command:

```
add rewrite policy pol_replace_http_with_https TRUE act_replace_http_with_https
 NOREWRITE
```

- Bind your new policy globally.

To troubleshoot this rewrite operation, see "Case Study: Rewrite Policy for Converting HTTP Links to HTTPS not Working."

Removing Unwanted Headers

This example explains how to use a Rewrite policy to remove unwanted headers. Specifically, the example shows how to remove the following headers:

- **Accept Encoding header.** Removing the Accept Encoding header from HTTP responses prevents compression of the response.
- **Content Location header.** Removing the Content Location header from HTTP responses prevents your server from providing a hacker with information that might allow a security breach.

To delete headers from HTTP responses, you create a rewrite action and a rewrite policy, and you bind the policy globally.

To create the appropriate Rewrite action by using the command line interface

At the command prompt, type one of the following commands to either remove the Accept Encoding header and prevent response compression or remove the Content Location header:

- `add rewrite action "act_remove-ae"delete_http_header "Accept-Encoding"`
- `add rewrite action "act_remove-cl"delete_http_header "Content-Location"`

To create the appropriate Rewrite policy by using the command line interface

At the command prompt, type one of the following commands to remove either the Accept Encoding header or the Content Location header:

- `add rewrite policy "pol_remove-ae"true "act_remove-ae"`
- `add rewrite policy "pol_remove-cl"true "act_remove-cl"`

To bind the policy globally by using the command line interface

At the command prompt, type one of the following commands, as appropriate, to globally bind the policy that you have created:

- `bind rewrite global pol_remove_ae 100`
- `bind rewrite global pol_remove_cl 200`

Reducing Web Server Redirects

This example explains how to use a Rewrite policy to modify connections to your home page and other URLs that end with a forward slash (/) to the default index page for your server, preventing redirects and reducing load on your server.

To modify directory-level HTTP requests to include the default home page by using the command line

- To create a Rewrite action named action-default-homepage that modifies URLs that end in a forward slash to include the default home page index.html, type:

```
add rewrite action "action-default-homepage"replace q##http.req.url.path "/
""/index.html"##
```

- To create a Rewrite policy named policy-default-homepage that detects connections to your home page and applies your new action, type:

```
add rewrite policy "policy-default-homepage"q##http.req.url.path.EQ("/")"
action-default-homepage"##
```

- Globally bind your new policy to put it into effect.

Masking the Server Header

This example explains how to use a Rewrite policy to mask the information in the Server header in HTTP responses from your Web server. That header contains information that hackers can use to compromise your Web site. While masking the header will not prevent a skilled hacker from finding out information about your server, it will make hacking your Web server more difficult and encourage hackers to choose less well protected targets.

To mask the Server header in responses from the command line

1. To create a Rewrite action named act_mask-server that replaces the contents of the Server header with an uninformative string, type:

```
add rewrite action "act_mask-server"replace "http.RES.HEADER("Server")"""
Web Server 1.0""
```

1. To create a Rewrite policy named pol_mask-server that detects all connections, type:

```
add rewrite policy "pol_mask-server"true "act_mask-server"
```

1. Globally bind your new policy to put it into effect.

Tutorial Examples of Classic Policies

September 21, 2020

The following examples describe useful examples of classic policy configuration for certain NetScaler features, such as NetScaler Gateway, application firewall, and SSL.

This document includes the following details:

- NetScaler Gateway Policy to Check for a Valid Client Certificate
- Application Firewall Policy to Protect a Shopping Cart Application
- Application Firewall Policy to Protect Scripted Web Pages
- DNS Policy to Drop Packets from Specific IPs
- SSL Policy to Require Valid Client Certificates

NetScaler Gateway Policy to Check for a Valid Client Certificate

The following policies enable the NetScaler to ensure that a client presents a valid certificate before establishing a connection to a company's SSL VPN.

To check for a valid client certificate by using the command line interface

- Add an action to perform client certificate authentication.

  ```
  add ssl action act1 -clientAuth DOCLIENTAUTH
  ```

- Create an SSL policy to evaluate the client requests.

  ```
  add ssl policy pol1 -rule "REQ.HTTP.METHOD == GET"-action act1
  ```

- Add a rewrite action to insert the certificate issuer details into the HTTP header of the requests being sent to web server.

  ```
  add rewrite action act2 insert\\_http\\_header "CertDN"CLIENT.SSL.
  CLIENT\\_CERT.SUBJECT
  ```

- Create a rewrite policy to insert the certificate issuer details, if the client certificate exists.

  ```
  add rewrite policy pol2 "CLIENT.SSL.CLIENT\\_CERT.EXISTS"act2
  ```

  Bind these new policies to the NetScaler VIP to put them into effect.

Application firewall policy to protect a shopping cart application

Shopping cart applications handle sensitive customer information, for example, credit card numbers and expiration dates, and they access back-end database servers. Many shopping cart applications also use legacy CGI scripts, which can contain security flaws that were unknown at the time they were written, but are now known to hackers and identity thieves.

A shopping cart application is particularly vulnerable to the following attacks:

- **Cookie tampering.** If a shopping cart application uses cookies, and does not perform the appropriate checks on the cookies that users return to the application, an attacker could modify a cookie and gain access to the shopping cart application under another user's credentials. Once logged on as that user, the attacker could obtain sensitive private information about the legitimate user or place orders using the legitimate user's account.
- **SQL injection.** A shopping cart application normally accesses a back-end database server. Unless the application performs the appropriate safety checks on the data users return in the form fields of its Web forms before it passes that information on to the SQL database, an attacker can use a Web form to inject unauthorized SQL commands into the database server. Attackers normally use this type of attack to obtain sensitive private information from the database or modify information in the database.

The following configuration will protect a shopping cart application against these and other attacks.

To protect a shopping cart application by using the GUI

1. Navigate to **Security > Application Firewall > Profiles**, and then click **Add.**
2. In the **Create Application Firewall Profile** dialog box, in the Profile Name field, enter shopping_cart.
3. In the Profile Type drop-down list, select Web Application.
4. In the Configure Select Advanced defaults.
5. Click **Create** and then click **Close.**
6. In the details view, double-click the new profile.
7. In the Configure Web Application Profile dialog box, configure your new profile as described below:
   - Click the Checks tab, double-click the Start URL check, and in the Modify Start URL Check dialog box, click the General tab and disable blocking, and enable learning, logging, statistics, and URL closure. Click OK and then click Close.

     Note that if you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

     `set appfw profile shopping\\_cart -startURLAction LEARN LOG STATS -startURLClosure ON`

   - For the Cookie Consistency check and Form Field Consistency checks, disable blocking, and enable learning, logging, statistics, using a similar method to the Modify Start URL Check configuration.

     If you are using the command line, you configure these settings by typing the following commands:

     `set appfw profile shopping\\_cart -cookieConsistencyAction LEARN LOG STATS`

     `set appfw profile shopping\\_cart -fieldConsistencyAction LEARN LOG STATS`

   - For the SQL Injection check, disable blocking, and enable learning, logging, statistics, and transformation of special characters in the Modify SQL Injection Check dialog box, General tab, Check Actions section.

     If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

```
set appfw profile shopping\\_cart -SQLInjectionAction LEARN LOG STATS -SQLInjectionTransformSpecialChars ON
```

- For the Credit Card check, disable blocking; enable logging, statistics, and masking of credit card numbers; and enable protection for those credit cards you accept as forms of payment.
  - If you are using the GUI, you configure blocking, logging, statistics, and masking (or x-out) in the Modify Credit Card Check dialog box, General tab, Check Actions section. You configure protection for specific credit cards in the Settings tab of the same dialog box.
  - If you are using the command line, you configure these settings by typing the following at the prompt, and pressing ENTER:

  ```
  set appfw profile shopping_cart -creditCardAction LOG STATS -creditCardXOut ON -creditCard <name> [<name>...]
  ```

  For <name> you substitute the name of the credit card you want to protect. For Visa, you substitute VISA. For Master Card, you substitute MasterCard. For American Express, you substitute Amex. For Discover, you substitute Discover. For Diners Club, you substitute DinersClub. For JCB, you substitute JCB.

8. Create a policy named shopping_cart that detects connections to your shopping cart application and applies the shopping_cart profile to those connections.

   To detect connections to the shopping cart, you examine the URL of incoming connections. If you host your shopping cart application on a separate host (a wise measure for security and other reasons), you can simply look for the presence of that host in the URL. If you host your shopping cart in a directory on a host that handles other traffic, as well, you must determine that the connection is going to the appropriate directory and/or HTML page.

   The process for detecting either of these is the same; you create a policy based on the following expression, and substitute the proper host or URL for `<string>`.

   ```
   REQ.HTTP.HEADER URL CONTAINS <string>
   <!--NeedCopy-->
   ```

   - If you are using the GUI, you navigate to the application firewall Policies page, click the Add… button to add a new policy, and follow the policy creation process described in "To create a policy with classic expressions using the GUI" beginning on page 201 and following.

   - If you are using the command line, you type the following command at the prompt and press Enter:

```
add appfw policy shopping_cart "REQ.HTTP.HEADER URL CONTAINS <
string>"shopping_cart
```

2. Globally bind your new policy to put it into effect.

Because you want to ensure that this policy will match all connections to the shopping cart, and not be preempted by another more general policy, you should assign a high priority to it. If you assign one (1) as the priority, no other policy can preempt this one.

Application firewall policy to protect scripted web pages

Web pages with embedded scripts, especially legacy JavaScripts, often violate the "same origin rule," which does not allow scripts to access or modify content on any server but the server where they are located. This security vulnerability is called cross-site scripting. The application firewall Cross-Site Scripting rule normally filters out requests that contain cross-site scripting.

Unfortunately, this can cause Web pages with older JavaScripts to stop functioning, even when your system administrator has checked those scripts and knows that they are safe. The example below explains how to configure the application firewall to allow cross-site scripting in Web pages from trusted sources without disabling this important filter for the rest of your Web sites.

To protect Web pages with cross-site scripting by using the command line interface

- At the command line, to create an advanced profile, type:

  ```
  add appfw profile pr_xssokay -defaults advanced
  ```

- To configure the profile, type:

  ```
  set appfw profile pr_xssokay -startURLAction NONE -startURLClosure OFF
  -cookieConsistencyAction LEARN LOG STATS -fieldConsistencyAction LEARN
  LOG STATS -crossSiteScriptingAction LEARN LOG STATS$"
  ```

- Create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile, type:

  ```
  add appfw policy pol_xssokay "REQ.HTTP.HEADER URL CONTAINS ^\\.pl\\?$
  || REQ.HTTP.HEADER URL CONTAINS ^\\.js$"pr\\_xssokay
  ```

- Globally bind the policy.

To protect Web pages with cross-site scripting by using the GUI

1. Navigate to **Security > Application Firewall > Profiles.**
2. In the details view, click **Add.**

3. In the **Create Application Firewall Profile** dialog box, create a Web Application profile with advanced defaults and name it pr_xssokay. Click **Create** and then click **Close.**
4. In the details view, click the profile, click Open, and in the Configure Web Application Profile dialog box, configure the pr_xssokay profile as shown below.

   Start URL Check: Clear all actions.

   - Cookie Consistency Check: Disable blocking.
   - Form Field Consistency Check: Disable blocking.
   - Cross-Site Scripting Check: Disable blocking.

   This should prevent blocking of legitimate requests involving Web pages with cross-site scripting that you know are nonetheless safe.
5. Click **Policies**, and then click **Add.**
6. In the **Create Application Firewall Policy** dialog box, create a policy that detects connections to your scripted Web pages and applies the pr_xssokay profile:

   - Policy name: pol_xssokay
   - Associated profile: pr_xssokay

Policy expression: “REQ.HTTP.HEADER URL CONTAINS ^\.pl\?$	REQ.HTTP.HEADER URL CONTAINS ^\.js$”

7. Globally bind your new policy to put it into effect.

DNS Policy to drop packets from specific IPs

The following example describes how to create a DNS action and DNS policy that detects connections from unwanted IPs or networks, such as those used in a DDOS attack, and drops all packets from those locations. The example shows networks within the IANA reserved IP block 192.168.0.0/16. A hostile network will normally be on publicly routable IPs.

To drop packets from specific IPs by using the command line interface

- To create a DNS policy named pol_ddos_drop that detects connections from hostile networks and drops those packets, type:

  ```
  add dns policy pol_ddos_drop 'client.ip.src.in_subnet(192.168.253.128/25)
  || client.ip.src.in_subnet(192.168.254.32/27)'-drop YES'
  ```

For the example networks in the 192.168.0.0/16 range, you substitute the IP and netmask in ###.###.###.###/## format of each network you want to block. You can include as many networks as you want, separating each CLIENT.IP.SRC.IN_SUBNET(###.###.###.###./##) command with the OR operator.

- Globally bind your new policy to put it into effect.

SSL policy to require valid client certificates

The following example shows an SSL policy that checks the user's client certificate validity before initiating an SSL connection with a client.

To block connections from users with expired client certificates

- Log on to the command line interface.

  If you are using the GUI, navigate to the SSL Policies page, then in the Data area, click the Actions tab.

- Create an SSL action named act_current_client_cert that requires that users have a current client certificate to establish an SSL connection with the NetScaler.

  ```
  add ssl action act_current_client_cert-clientAuth DOCLIENTAUTH -clientCert
   ENABLED -certHeader "clientCertificateHeader"-clientCertNotBefore
  ENABLED -certNotBeforeHeader "Mon, 01 Jan 2007 00:00:00 GMT"
  ```

- Create an SSL policy named pol_current_client_cert that detects connections to the Web server that contain a query string.

  ```
  add ssl policy pol_current_ client_cert 'REQ.SSL.CLIENT.CERT.VALIDFROM
  \>= "Mon, 01 Jan 2007 00:00:00 GMT"'act_block_ssl
  ```

- Bind your new policy globally.

  Because this SSL policy should apply to any user's SSL connection unless a more specific SSL policy applies, you may want to assign it a low priority. If you assign it a priority of one thousand (1000), that should ensure that other SSL policies are evaluated first, meaning that this policy will apply only to connections that do not match more specific policy criteria.

Migrate Apache mod_rewrite rules to default policy syntax

April 14, 2021

The Apache HTTP Server provides an engine known as mod_rewrite for rewriting HTTP request URLs. If you migrate the mod_rewrite rules from Apache to the NetScaler, you boost back-end server performance. In addition, because the NetScaler typically load balances multiple (sometimes thousands of) Web servers, after migrating the rules to the NetScaler you have a single point of control for these rules.

Following are examples of mod_rewrite functions, and translations of these functions into the Rewrite and Responder policies on the NetScaler.

Convert URL variations into canonical URLs

For some Web servers you can have multiple URLs for a resource. Although the canonical URLs must be used and distributed, other URLs can exist as shortcuts or internal URLs. You can make sure that users see the canonical URL regardless of the URL used to make an initial request.

In the following examples, the URL `/~user` is converted to `/u/user`.

Apache mod_rewrite solution for converting a URL

```
RewriteRule ^/~([^/]+)/?(.*) /u/$1/$2[R]
<!--NeedCopy-->
```

Solution for converting a URL

```
add responder action act1 redirect '"/u/"+HTTP.REQ.URL.AFTER_STR("/~")'
    -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.STARTSWITH("/~") && HTTP.REQ.
   URL.LENGTH.GT(2)' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Convert host name variations to canonical host names

You can enforce the use of a particular host name for reaching a site. For example, you can enforce the use of www.example.com instead of example.com.

Apache mod_rewrite solution for enforcing a particular host name for sites running on a port other than 80

```
RewriteCond %{
 HTTP_HOST }
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
RewriteCond %{
 SERVER_PORT }
  !^80$
RewriteRule ^/(.*)         http://www.example.com:%{
 SERVER_PORT }
 /$1 [L,R]
<!--NeedCopy-->
```

Apache mod_rewrite solution for enforcing a particular host name for sites running on port 80

```
RewriteCond %{
 HTTP_HOST }
    !^www.example.com
RewriteCond %{
 HTTP_HOST }
    !^$
RewriteRule ^/(.*)         http://www.example.com/$1 [L,R]
<!--NeedCopy-->
```

Solution for enforcing a particular host name for sites running on a port other than 80

```
add responder action act1 redirect '"http://www.example.com:"+CLIENT.TCP.DSTPORT+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME.EQ("")&&!HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Solution for enforcing a particular host name for sites running on port 80

```
add responder action act1 redirect '"http://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1  '!HTTP.REQ.HOSTNAME.CONTAINS("www.example.com")&&!HTTP.REQ.HOSTNAME.EQ("")&&HTTP.REQ.HOSTNAME.PORT.EQ(80)&&HTTP.REQ.HOSTNAME.CONTAINS("example.com")' act1
bind responder global  pol1 100 END
<!--NeedCopy-->
```

Move a document root

Usually the document root of a Web server is based on the URL "/". However, the document root can be any directory. You can redirect traffic to the document root if it changes from the top-level "/" directory to another directory.

In the following examples, you change the document root from / to /e/www. The first two examples simply replace one string with another. The third example is more universal because, along with replacing the root directory, it preserves the rest of the URL (the path and query string), for example, redirecting /example/file.html to /e/www/example/file.html.

Apache mod_rewrite solution for moving the document root

```
RewriteEngine on
RewriteRule   ^/$  /e/www/  [R]
<!--NeedCopy-->
```

Solution for moving the document root

```
add responder action act1 redirect '"/e/www/"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.EQ("/")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Solution for moving document root and appending path information to the request

```
add responder action act1 redirect '"/e/www"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.URL.STARTSWITH("/e/www/")' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Move home directories to a new web server

You might want to redirect requests that are sent to home directories on a Web server to a different Web server. For example, if a new Web server is replacing an old one over time, as you migrate home directories to the new location you need to redirect requests for the migrated home directories to the new Web server.

In the following examples, the host name for the new Web server is `newserver`.

Apache mod_rewrite solution for redirecting to another Web server

```
RewriteRule   ^/(.+)  http://newserver/$1     [R,L]
<!--NeedCopy-->
```

NetScaler solution for redirecting to another Web server (method 1)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.REGEX_MATCH(re#^/(.+)#)'   act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Solution for redirecting to another Web server (method 2)

```
add responder  action act1 redirect '"http://newserver"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.LENGTH.GT(1)' act1
bind responder global pol1 100 END
<!--NeedCopy-->
```

Working with structured home directories

Typically, a site with thousands of users has a structured home directory layout. For example, each home directory might reside under a subdirectory that is named using the first character of the user name. For example, the home directory for jsmith (/~jsmith/anypath) might be `/home/j/smith/.www/anypath`, and the home directory for `rvalveti (/~rvalveti/anypath)` might be `/home/r/rvalveti/.www/anypath`.

The following examples redirect requests to the home directory.

Apache mod_rewrite solution for structured home directories

```
RewriteRule   ^/~(([a-z])[a-z0-9]+)(.*)  /home/$2/$1/.www$3
<!--NeedCopy-->
```

Solution for structured home directories

```
NetScaler solution for structured home directories

add rewrite action act1 replace 'HTTP.REQ.URL'  '"/home/"+ HTTP.REQ.URL.AFTER_STR("~").PREFIX(1)+"/"+ HTTP.REQ.URL.AFTER_STR("~").BEFORE_STR("/")+"/.www"+HTTP.REQ.URL.SKIP(\'/\',1)'  -bypassSafetyCheck yes
add rewrite policy pol1  'HTTP.REQ.URL.PATH.STARTSWITH("/~")' act1
bind rewrite global pol1 100
<!--NeedCopy-->
```

Redirecting invalid URLs to other web servers

If a URL is not valid, it must be redirected to must Web server. For example, you must redirect to another Web server if a file that is named in a URL does not exist on the server that is named in the URL.

On Apache, you can perform this check using mod_rewrite. On the NetScaler, an HTTP callout can check for a file on a server by running a script on the server. In the following NetScaler examples, a script named file_check.cgi processes the URL and uses this information to check for the presence of the target file on the server. The script returns TRUE or FALSE, and the NetScaler uses the value that the script returns to validate the policy.

In addition to performing the redirection, the NetScaler can add custom headers or, as in the second NetScaler example, it can add text in the response body.

Apache mod_rewrite solution for redirection if a URL is wrong

```
RewriteCond   /your/docroot/%{
 REQUEST_FILENAME }
  !-f
RewriteRule   ^(.+)      http://webserverB.com/$1 [R]
<!--NeedCopy-->
```

NetScaler solution for redirection if a URL is wrong (method 1)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -parameters query=http.req.url.path -headers Name("ddd")
add responder action act1 redirect '"http://webserverB.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
<!--NeedCopy-->
```

NetScaler solution for redirection if a URL is wrong (method 2)

```
add HTTPCallout Call
set policy httpCallout Call -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -parameters query=http.req.url.path -headers Name("ddd")
add responder  action act1 respondwith  '"HTTP/1.1 302 Moved Temporarily\r\nLocation: http://webserverB.com"+HTTP.REQ.URL+"\r\n\r\nHTTPCallout Used"' -bypassSafetyCheck yes
add responder policy pol1 '!HTTP.REQ.HEADER("Name").EXISTS  &&  !SYS.HTTP_CALLOUT(call)' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Rewriting a URL based on time

You can rewrite a URL based on the time. The following examples change a request for example.html to example.day.html or example.night.html, depending on the time of day.

Apache mod_rewrite solution for rewriting a URL based on the time

```
RewriteCond   %{
 TIME_HOUR }
 %{
 TIME_MIN }
  >0700
RewriteCond   %{
 TIME_HOUR }
 %{
 TIME_MIN }
  <1900
RewriteRule   ^example\.html$ example.day.html [L]
RewriteRule   ^example\.html$ example.night.html
<!--NeedCopy-->
```

NetScaler solution for rewriting a URL based on the time

```
add rewrite action act1 insert_before 'HTTP.REQ.URL.PATH.SUFFIX
    (\'.\',0)' '"day."'
add rewrite action act2  insert_before 'HTTP.REQ.URL.PATH.SUFFIX
    (\'.\',0)' '"night."'
add rewrite  policy pol1 'SYS.TIME.WITHIN(LOCAL 07h 00m,LOCAL 18h 59m)'
     act1
add rewrite policy pol2 'true'  act2
bind rewrite global pol1 101
bind rewrite global pol2 102
<!--NeedCopy-->
```

Redirecting to a new file name (Invisible to the User)

If you rename a webpage, you can continue to support the old URL for backward compatibility while preventing users from recognizing that the page was renamed.

In the first two of the following examples, the base directory is `/~example/`. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for managing a file name change in a fixed location

```
RewriteEngine  on
RewriteBase    /~example/
RewriteRule    ^foo\.html$  bar.html
<!--NeedCopy-->
```

NetScaler solution for managing a file name change in a fixed location

```
add rewrite action act1 replace 'HTTP.REQ.URL.AFTER_STR("/~example").
    SUBSTR("foo.html")' '"bar.html"'
```

```
add rewrite policy pol1 'HTTP.REQ.URL.ENDSWITH("/~example/foo.html")' act1
bind rewrite global pol1 100
<!--NeedCopy-->
```

Solution for file name change regardless of the base directory or query strings in the URL

```
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"bar.html"'
Add rewrite policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("foo.html")' act1
Bind rewrite global pol1 100
<!--NeedCopy-->
```

Redirecting to new file name (user-visible URL)

If you rename a webpage, you might want to continue to support the old URL for backward compatibility and allow users to see that the page was renamed by changing the URL that is displayed in the browser.

In the first two of the following examples, redirection occurs when the base directory is `/~example/`. The third example accommodates any base directory and the presence of query strings in the URL.

Apache mod_rewrite solution for changing the file name and the URL displayed in the browser

```
RewriteEngine on
RewriteBase    /~example/
RewriteRule    ^old\.html$ new.html   [R]
<!--NeedCopy-->
```

NetScaler solution for changing the file name and the URL displayed in the browser

```
add responder action act1 redirect 'HTTP.REQ.URL.BEFORE_STR("foo.html")+"new.html"' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.ENDSWITH("/~example/old.html")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

How to change file name and URL displayed in the browser regardless of the base directory or query strings in the URL

```
add responder action act1 redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("old.html")+"new.html"+HTTP.REQ.URL.AFTER_STR("old.html")' -bypassSafetyCheck yes
add responder policy pol1 'HTTP.REQ.URL.PATH.CONTAINS("old.html")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Accommodating browser dependent content

To accommodate browser-specific limitations—at least for important top-level pages—it is sometimes necessary to set restrictions on the browser type and version. For example, you might want to set a maximum version for the latest Netscape variants, a minimum version for Lynx browsers, and an average feature version for all others.

The following examples act on the HTTP header "User-Agent", such that if this header begins with "Mozilla/3", the page `MyPage.html` is rewritten to `MyPage.NS.html`. If the browser is "Lynx" or "Mozilla" version 1 or 2, the URL becomes `MyPage.20.html`. All other browsers receive the page `MyPage.32.html`.

Apache mod_rewrite solution for browser-specific settings

```
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/3.*
RewriteRule ^MyPage\.html$ MyPage.NS.html [L]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Lynx/.* [OR]
RewriteCond %{
 HTTP_USER_AGENT }
   ^Mozilla/[12].*
RewriteRule ^MyPage\.html$ MyPage.20.html [L]
RewriteRule ^fMyPage\.html$ MyPage.32.html [L]
NetScaler solution for browser-specific settings
add patset pat1
bind patset pat1 Mozilla/1
bind Patset pat1 Mozilla/2
bind patset pat1 Lynx
bind Patset pat1 Mozilla/3
add rewrite action act1 insert_before 'HTTP.REQ.URL.SUFFIX' '"NS."'
add rewrite action act2 insert_before 'HTTP.REQ.URL.SUFFIX' '"20."'
add rewrite action act3 insert_before 'HTTP.REQ.URL.SUFFIX' '"32."'
add rewrite policy pol1 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").EQ(4)' act1
add rewrite policy pol2 'HTTP.REQ.HEADER("User-Agent").STARTSWITH_INDEX("pat1").BETWEEN(1,3)' act2
add rewrite policy pol3 '!HTTP.REQ.HEADER("User-Agent").STARTSWITH_ANY("pat1")' act3
bind rewrite global pol1 101 END
bind rewrite global pol2 102 END
bind rewrite global pol3 103 END
<!--NeedCopy-->
```

Blocking access by robots

You can block a robot from retrieving pages from a specific directory or a set of directories to ease up the traffic to and from these directories. You can restrict access based on the specific location or you can block requests based on information in User-Agent HTTP headers.

In the following examples, the Web location to be blocked is `/~example/foo/arc/`, the IP addresses to be blocked are 1.1.1.1 and 10.10.10.10, and the robot's name is `NameOfBadRobot`.

Apache mod_rewrite solution for blocking a path and a User-Agent header

```
RewriteCond %{
 HTTP_USER_AGENT }
    ^NameOfBadRobot.*
RewriteCond %{
 REMOTE_ADDR }
        ^123\.45\.67\.[8-9]$
RewriteRule ^/~example/foo/arc/.+   -   [F]
<!--NeedCopy-->
```

NetScaler solution for blocking a path and a User-Agent header

```
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"'
add responder policy pol1 'HTTP.REQ.HEADER("User_Agent").STARTSWITH("NameOfBadRobot")&&CLIENT.IP.SRC.EQ(123.45.67.8)&&CLIENT.IP.SRC.EQ(123.45.67.9) && HTTP.REQ.URL.STARTSWITH("/~example/foo/arc")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Block access to inline images

If you find people frequently going to your server to copy inline graphics for their own use (and generating unnecessary traffic), you may want to restrict the browser's ability to send an HTTP Referee header.

Apache mod_rewrite solution for blocking access to an inline image

```
RewriteCond %{
 HTTP_REFERER }
  !^$
RewriteCond %{
 HTTP_REFERER }
  !^ttp://www.example.com/~graphics/.*$
RewriteRule .*\.png$ - [F]
<!--NeedCopy-->
```

NetScaler solution for blocking access to an inline image

```
add patset pat1
bind patset pat1 .png
bind patset pat1 .jpeg
add responder action act1 respondwith '"HTTP/1.1 403 Forbidden\r\n\r\n"
    '
add responder policy pol1 '!HTTP.REQ.HEADER("Referer").EQ("") && !HTTP.
    REQ.HEADER("Referer").STARTSWITH("ttp://www.example.com/~graphics/")
    &&HTTP.REQ.URL.ENDSWITH_ANY("pat1")' act1
bind responder global pol1 100
<!--NeedCopy-->
```

Creating extensionless links

To prevent users from knowing application or script details on the server side, you can hide file name extensions from users. To do this, you may want to support extensionless links. You can achieve this behavior by using rewrite rules to add an extension to all requests, or to selectively add extensions to requests.

The first two of the following examples show adding an extension to all request URLs. In the last example, one of two file name extensions is added. In the last example, the mod_rewrite module can easily find the file name extension because this module resides on the Web server. In contrast, the NetScaler must invoke an HTTP callout to check the extension of the requested file on the Web server. Based on the callout response, the NetScaler adds the .html or .php extension to the request URL.

> **Note:**
>
> In the second NetScaler example, an HTTP callout is used to query a script named file_check.cgi hosted on the server. This script checks whether the argument that is provided in the callout is a valid file name.

Apache mod_rewrite solution for adding a .php extension to all requests

```
RewriteRule ^/?([a-z]+)$ $1.php [L]
<!--NeedCopy-->
```

NetScaler policy for adding a .php extension to all requests

```
add rewrite action act1 insert_after 'HTTP.REQ.URL' '".php"'
add rewrite policy pol1 'HTTP.REQ.URL.PATH.REGEX_MATCH(re#^/([a-z]+)$#)' act1
bind rewrite global pol1 100
<!--NeedCopy-->
```

Apache mod_rewrite solution for adding either .html or .php extensions to requests

```
RewriteCond %{
 REQUEST_FILENAME }
 .php -f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.php [L]
RewriteCond %{
 REQUEST_FILENAME }
 .html  - f
RewriteRule ^/?([a-zA-Z0-9]+)$ $1.html [L]
<!--NeedCopy-->
```

NetScaler policy for adding either .html or .php extensions to requests

```
add HTTPCallout Call_html
add HTTPCallout Call_php
set policy httpCallout Call_html -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"'   -parameters query=http.req.url+".html"
set policy httpCallout Call_php -IPAddress 10.102.59.101 -port 80 -hostExpr '"10.102.59.101"' -returnType BOOL -ResultExpr 'HTTP.RES.BODY(100).CONTAINS("True")'  -urlStemExpr '"/cgi-bin/file_check.cgi"' -parameters query=http.req.url+".php"
add patset pat1
bind patset pat1 .html
bind patset pat1 .php
bind patset pat1 .asp
bind patset pat1 .cgi
add rewrite  action act1 insert_after 'HTTP.REQ.URL.PATH'  '".html"'
add rewrite  action act2 insert_after "HTTP.REQ.URL.PATH"  '".php"'
add rewrite policy pol1 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_html)' act1
add rewrite policy pol2 '!HTTP.REQ.URL.CONTAINS_ANY("pat1") && SYS.HTTP_CALLOUT(Call_php)' act2
bind rewrite global pol1 100 END
bind rewrite global pol2 101 END
<!--NeedCopy-->
```

Redirecting a Working URI to a New Format

Suppose that you have a set of working URLs that resemble the following:

```
/index.php?id=nnnn
<!--NeedCopy-->
```

To change these URLs to `/nnnn` and make sure that search engines update their indexes to the new URI format, you need to do the following:

- Redirect the old URIs to the new ones so that search engines update their indexes.
- Rewrite the new URI back to the old one so that the `index.php` script runs correctly.

To accomplish this, you can insert marker code into the query string (making sure that the marker code is not seen by the user), and then removing the marker code for the index.php script.

The following examples redirect from an old link to a new format only if a marker is not present in the query string. The link that uses the new format is rewritten back to the old format, and a marker is added to the query string.

Apache mod_rewrite solution

```
RewriteCond %{
 QUERY_STRING }
  !marker
```

```
RewriteCond %{
 QUERY_STRING }
  id=([-a-zA-Z0-9_+]+)
RewriteRule ^/?index\.php$ %1? [R,L]
RewriteRule ^/?([-a-zA-Z0-9_+]+)$  index.php?marker&id=$1 [L]
NetScaler solution
add responder action act_redirect redirect 'HTTP.REQ.URL.PATH.BEFORE_STR("index.php")+HTTP.REQ.URL.QUERY.VALUE("id")' -bypassSafetyCheck yes
add responder policy pol_redirect '!HTTP.REQ.URL.QUERY.CONTAINS("marker")&& HTTP.REQ.URL.QUERY.VALUE("id").REGEX_MATCH(re/[-a-zA-Z0-9_+]+/) && HTTP.REQ.URL.PATH.CONTAINS("index.php")' act_redirect
bind responder global pol_redirect 100 END
add rewrite action act1 replace 'HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' '"index.phpmarker&id="+HTTP.REQ.URL.PATH.SUFFIX(\'/\',0)' -bypassSafetyCheck yes
add rewrite policy pol1 '!HTTP.REQ.URL.QUERY.CONTAINS("marker")'  act1
bind rewrite global pol1 100 END
<!--NeedCopy-->
```

Ensuring That a Secure Server Is Used for Selected Pages

To make sure that only secure servers are used for selected webpages, you can use the following Apache mod_rewrite code or NetScaler Responder policies.

Apache mod_rewrite solution

```
RewriteCond %{
 SERVER_PORT }
  !^443$
RewriteRule ^/?(page1|page2|page3|page4|page5)$  https://www.example.com/%1 [R,L]
<!--NeedCopy-->
```

NetScaler solution using regular expressions

```
add responder action res_redirect redirect  '"https://www.example.com"+HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.REQ.URL.REGEX_MATCH(re/page[1-5]/)'  res_redirect
bind responder global pol_redirect 100 END
<!--NeedCopy-->
```

NetScaler solution using pattern sets

```
add patset pat1
bind patset pat1 page1
bind patset pat1 page2
bind patset pat1 page3
bind patset pat1 page4
bind patset pat1 page5
add responder action res_redirect redirect  '"https://www.example.com"+
    HTTP.REQ.URL' -bypassSafetyCheck yes
add responder policy pol_redirect '!CLIENT.TCP.DSTPORT.EQ(443)&&HTTP.
    REQ.URL.CONTAINS_ANY("pat1")'  res_redirect
bind responder global pol_redirect 100 END
<!--NeedCopy-->
```

Rate limiting

September 21, 2020

The rate limiting feature enables you to define the maximum load for a given network entity or virtual entity on the NetScaler appliance. The feature enables you to configure the appliance to monitor the rate of traffic associated with the entity and take preventive action, in real time, based on the traffic rate. This feature is particularly useful when the network is under attack from a hostile client that is sending the appliance a flood of requests. You can mitigate the risks that affect the availability of resources to clients, and you can improve the reliability of the network and the resources that the appliance manages.

You can monitor and control the rate of traffic that is associated with virtual and user-defined entities, including virtual servers, URLs, domains, and combinations of URLs and domains. You can throttle the rate of traffic if it is too high, base information caching on the traffic rate, and redirect traffic to a given load balancing virtual server if the traffic rate exceeds a predefined limit. You can apply rate-based monitoring to HTTP, TCP, and DNS requests.

To monitor the rate of traffic for a given scenario, you configure a *rate limit identifier*. A rate limit identifier specifies numeric thresholds such as the maximum number of requests or connections (of a particular type) that are permitted in a specified time period called a *time slice*.

Optionally, you can configure filters, known as *stream selectors*, and associate them with rate limit identifiers when you configure the identifiers. After you configure the optional stream selector and the limit identifier, you must invoke the limit identifier from a default syntax policy. You can invoke identifiers from any feature in which the identifier may be useful, including rewrite, responder, DNS, and integrated caching.

You can globally enable and disable SNMP traps for rate limit identifiers. Each trap contains cumulative data for the rate limit identifier's configured data collection interval (time slice), unless you specified multiple traps to be generated per time slice. For more information about configuring SNMP traps and managers, see "SNMP."

Configuring a Stream Selector

September 21, 2020

A traffic stream selector is an optional filter for identifying an entity for which you want to throttle access. The selector is applied to a request or a response and selects data points (keys) that can be analyzed by a rate stream identifier. These data points can be based on almost any characteristic of the traffic, including IP addresses, subnets, domain names, TCP or UDP identifiers, and particular strings or extensions in URLs.

A stream selector consists of individual default syntax expressions called selectlets. Each selectlet is a non-compound default syntax expression. A traffic stream selector can contain up to five non-compound expressions called selectlets. Each selectlet is considered to be in an AND relationship with the other expressions. Following are some examples of selectlets:

```
http.req.url
http.res.body(1000>after_str("car_model").before_str("made_in")"
"client.ip.src.subnet(24)"
<!--NeedCopy-->
```

The order in which you specify parameters is significant. For example, if you configure an IP address and a domain (in that order) in one selector, and then specify the domain and the IP address (in the reverse order) in another selector, the NetScaler considers these values to be unique. This can lead to the same transaction being counted twice. Also, if multiple policies invoke the same selector, the NetScaler, again, can count the same transaction more than once.

Note: If you modify an expression in a stream selector, you may get an error if any policy that invokes it is bound to a new policy label or bind point. For example, suppose that you create a stream selector named myStreamSelector1, invoke it from myLimitID1, and invoke the identifier from a DNS policy named dnsRateLimit1. If you change the expression in myStreamSelector1, you might receive an error when binding dnsRateLimit1 to a new bind point. The workaround is to modify these expressions before creating the policies that invoke them.

To configure a traffic stream selector by using the command line interface

At the command prompt, type:

```
add stream selector <name> <rule> ...
<!--NeedCopy-->
```

Example:

```
add stream selector myStreamSel HTTP.REQ.URL CLIENT.IP.SRC
<!--NeedCopy-->
```

To configure a stream selector by using the GUI

Navigate to AppExpert > Rate Limiting > Selectors, click Add and specify the relevant details.

Configuring a traffic rate limit identifier

September 21, 2020

A rate limit identifier checks if the amount of traffic exceeds a specified value, within a particular time interval. The identifier returns a "Boolean TRUE" if the amount of traffic exceeds a limit within a particular time interval. When you include a limit identifier in the compound default syntax expression in a policy rule, you must include a stream selector. If you do not specify, the limit identifier is applied to all requests or responses identified by the compound expression.

> **Note:**
>
> The maximum length for storing string results (for example, HTTP.REQ.URL) is 60 characters. If the string (for example, URL) is 1000 characters long, out of which 50 characters are long enough to uniquely identify a string, you can use an expression to extract required 50 characters.

To configure a traffic limit identifier from the command line interface

At the command prompt, type:

```
add ns limitIdentifier <limitIdentifier> -threshold <positive_integer> -timeSlice <positive_integer> -mode <mode> -limitType ( BURSTY | SMOOTH ) -selectorName <string> -maxBandwidth <positive_integer> -trapsInTimeSlice <positive_integer>
<!--NeedCopy-->
```

Argument description

limitIdentifier. Name for a rate limit identifier. Must begin with an ASCII letter or underscore (_) character, and must consist only of ASCII alphanumeric or underscore characters. Reserved words must not be used. This is a mandatory argument. Maximum Length: 31

threshold. A maximum number of requests that are allowed in the given timeslice when requests (mode is set as REQUEST_RATE) are tracked per timeslice. When connections (mode is set as CONNECTION) are tracked, it is the total number of connections that would be let through. Default value: 1 Minimum value: 1 Maximum Value: 4294967295

timeSlice. Time interval, in milliseconds, specified in multiples of 10, during which requests are tracked to check if they cross the threshold. This argument is needed only when the mode is set to REQUEST_RATE. Default value: 1000 Minimum value: 10 Maximum Value: 4294967295

mode. Defines the type of traffic to be tracked.
REQUEST_RATE - Tracks requests/timeslice.
CONNECTION - Tracks active transactions.

limitType. Smooth or bursty request type.

selectorName. Name of the rate limit selector. If this argument is NULL, rate limiting will be applied on all traffic received by the virtual server or the NetScaler (depending on whether the limit identifier is bound to a virtual server or globally) without any **filtering. Maximum Length: 31**

maxBandwidth. Maximum bandwidth permitted, in kbps. Minimum value: 0 Maximum value: 4294967287

Example:

Configuring traffic rate limit identifier in BURSTY mode:

```
add ns limitIdentifier 100_request_limit -threshold 100 -timeSlice 1000 -mode REQUEST_RATE -limitType BURSTY -selectorName limit_100_requests_selector -trapsInTimeSlice 30
<!--NeedCopy-->
```

Configuring traffic rate limit identifier in SMOOTH mode:

```
add ns limitidentifier limit_req -mode request_rate -limitType smooth -timeslice 1000 -Threshold 2000 -trapsInTimeSlice 200
<!--NeedCopy-->
```

To configure a traffic limit identifier by using the Citrix ADC GUI

Navigate to **AppExpert > Rate Limiting > Limit Identifiers**, click **Add** and specify the relevant details.

Configuring and Binding a Traffic Rate Policy

September 21, 2020

You implement rate-based application behavior by configuring a policy in an appropriate NetScaler feature. The feature must support default syntax policies. The policy expression must contain the following expression prefix to enable the feature to analyze the traffic rate:

```
sys.check_limit(<limit_identifier>)
<!--NeedCopy-->
```

Where limit_identifier is the name of a limit identifier.

The policy expression must be a compound expression that contains at least two components:

- An expression that identifies traffic to which the rate limit identifier is applied. For example:

```
    http.req.url.contains("my_aspx.aspx").
<!--NeedCopy-->
```

- An expression that identifies a rate limit identifier, for example, sys.check_limit("my_limit_identifier"). This must be the last expression in the policy expression.

To configure a rate-based policy by using the command line interface

At the command prompt, type the following command to configure a rate-based policy and verify the configuration:

```
add cache|dns|rewrite|responder policy <policy_name> -rule expression
    && sys.check_limit("<LimitIdentifierName>") [<feature-specific
    information>]
<!--NeedCopy-->
```

Following is a complete example of a rate-based policy rule. Note that this example assumes that you have configured the responder action, send_direct_url, that is associated with the policy. Note that the sys.check_limit parameter must be the last element of the policy expression:

```
add responder policy responder_threshold_policy "http.req.url.contains(
    "myindex.html") && sys.check_limit("my_limit_identifier")"
    send_direct_url
<!--NeedCopy-->
```

For information about binding a policy globally or to a virtual server, see "Binding Default Syntax Policies."

To configure a rate-based policy by using the GUI

1. In the navigation pane, expand the feature in which you want to configure a policy (for example, Integrated Caching, Rewrite, or Responder), and then click Policies.
2. In the details pane, click Add. In Name, enter a unique name for the policy.
3. Under Expression, enter the policy rule, and make sure that you include the sys.check_limit parameter as the final component of the expression. For example:

   ```
   http.req.url.contains("my_aspx.aspx") && sys.check_limit("
       my_limit_identifier")
   <!--NeedCopy-->
   ```

4. Enter feature-specific information about the policy.

   For example, you may be required to associate the policy with an action or a profile. For more information, see the feature-specific documentation.

5. Click Create, and then click Close.
6. Click Save.

Viewing the Traffic Rate

September 21, 2020

If traffic through one or more virtual servers matches a rate-based policy, you can view the rate of this traffic. The rate statistics are maintained in the limit identifier that you named in the rule for the rate-based policy. If more than one policy uses the same limit identifier, you can view the traffic rate as defined by hits to all of the policies that use the particular limit identifier.

To view the traffic rate by using the command line interface

At the command prompt, type the following command to view the traffic rate:

```
show ns limitSessions <limitIdentifier>
<!--NeedCopy-->
```

Example:

```
sh limitsession myLimitSession
<!--NeedCopy-->
```

To view the traffic rate by using the GUI

1. Navigate to AppExpert > Rate Limiting > Limit Identifiers.
2. Select a limit identifier whose traffic rate you want to view.
3. Click the Show Sessions button. If traffic through one or more virtual servers has matched a rate limiting policy that uses this limit identifier (and the hits are within the configured time slice for this identifier), the Session Details dialog box appears. Otherwise, you receive a "No session exists" message.

Testing a Rate-Based Policy

September 21, 2020

To test a rate-based policy, you can send traffic to any virtual server to which a rate-based policy is bound.

Task overview: Testing a rate-based policy

1. Configure a stream selector (optional) and a rate limit identifier (required). For example:

```
add stream selector sel_subnet Q.URL "CLIENT.IP.SRC.SUBNET(24)"
add ns limitIdentifier k_subnet -Threshold 4 -timeSlice 3600 -mode
     REQUEST_RATE  -limittype smooth  -selectorName sel_subnet  -
    trapsInTimeSlice 8
<!--NeedCopy-->
```

2. Configure the action that you want to associate with the policy that uses the rate limit identifier. For example:

```
add responder action resp_redirect redirect "\"http://
    response_site.com/\""
<!--NeedCopy-->
```

3. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier. For example, the policy can apply a rate limit identifier to all requests arriving from a particular subnet, as follows:

```
add responder policy resp_subnet "SYS.CHECK_LIMIT(\"k_subnet\")"
    resp_redirect
<!--NeedCopy-->
```

4. Bind the policy globally or to a virtual server. For example:

```
bind responder global resp_subnet 6 END -type DEFAULT
<!--NeedCopy-->
```

5. In a browser address bar, send a test HTTP query to a virtual server. For example:

```
http://<IP of a vserver>/testsite/test.txt
<!--NeedCopy-->
```

6. At the NetScaler command prompt, type:

```
show ns limitSessions \<limitIdentifier\>
<!--NeedCopy-->
```

Example

```
> sh limitsession k_subnet
1)      Time Remaining:       98 secs  Hits: 2
                                    Action Taken: 0
        Total Hash:    1718618  Hash String: /test.txt
        IPs gathered:
                1) 10.217.253.0
        Active Transactions: 0
 Done
>
<!--NeedCopy-->
```

7. Repeat the query and check the limit identifier statistics again to verify that the statistics are being updated correctly.

Examples of Rate-Based Policies

September 21, 2020

The following table shows examples of rate-based policies.

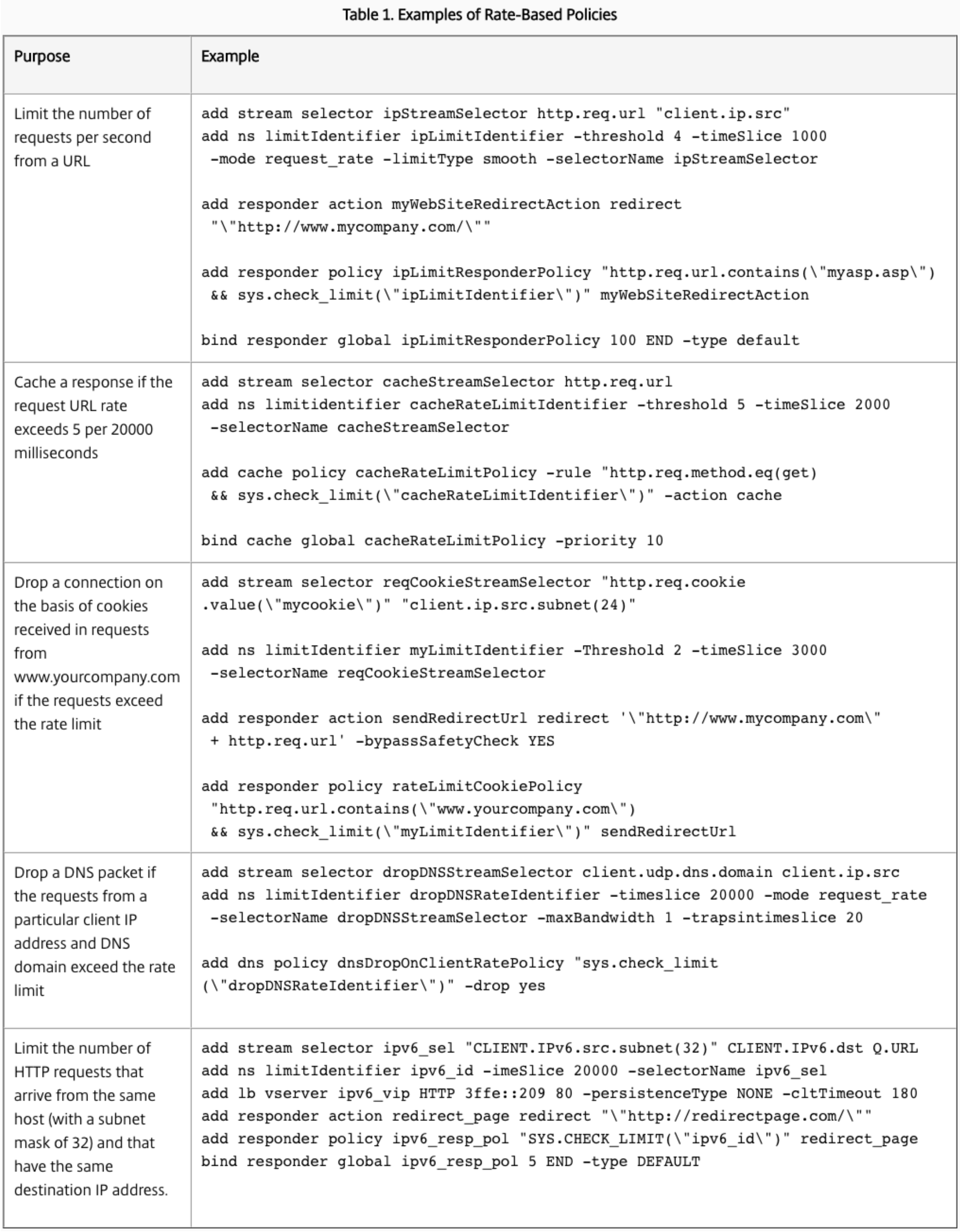


Table 1. Examples of Rate-Based Policies

Purpose	Example

Limit the number of requests per second from a URL

```
add stream selector ipStreamSelector http.req.url "client.ip.src"
add ns limitIdentifier ipLimitIdentifier -threshold 4 -timeSlice 1000
 -mode request_rate -limitType smooth -selectorName ipStreamSelector

add responder action myWebSiteRedirectAction redirect
 "\"http://www.mycompany.com/\""

add responder policy ipLimitResponderPolicy "http.req.url.contains(\"myasp.asp\")
 && sys.check_limit(\"ipLimitIdentifier\")" myWebSiteRedirectAction

bind responder global ipLimitResponderPolicy 100 END -type default
```

Cache a response if the request URL rate exceeds 5 per 20000 milliseconds

```
add stream selector cacheStreamSelector http.req.url
add ns limitidentifier cacheRateLimitIdentifier -threshold 5 -timeSlice 2000
 -selectorName cacheStreamSelector

add cache policy cacheRateLimitPolicy -rule "http.req.method.eq(get)
 && sys.check_limit(\"cacheRateLimitIdentifier\")" -action cache

bind cache global cacheRateLimitPolicy -priority 10
```

Drop a connection on the basis of cookies received in requests from www.yourcompany.com if the requests exceed the rate limit

```
add stream selector reqCookieStreamSelector "http.req.cookie
.value(\"mycookie\")" "client.ip.src.subnet(24)"

add ns limitIdentifier myLimitIdentifier -Threshold 2 -timeSlice 3000
 -selectorName reqCookieStreamSelector

add responder action sendRedirectUrl redirect '\"http://www.mycompany.com\"
 + http.req.url' -bypassSafetyCheck YES

add responder policy rateLimitCookiePolicy
 "http.req.url.contains(\"www.yourcompany.com\")
 && sys.check_limit(\"myLimitIdentifier\")" sendRedirectUrl
```

Drop a DNS packet if the requests from a particular client IP address and DNS domain exceed the rate limit

```
add stream selector dropDNSStreamSelector client.udp.dns.domain client.ip.src
add ns limitIdentifier dropDNSRateIdentifier -timeslice 20000 -mode request_rate
 -selectorName dropDNSStreamSelector -maxBandwidth 1 -trapsintimeslice 20

add dns policy dnsDropOnClientRatePolicy "sys.check_limit
(\"dropDNSRateIdentifier\")" -drop yes
```

Limit the number of HTTP requests that arrive from the same host (with a subnet mask of 32) and that have the same destination IP address.

```
add stream selector ipv6_sel "CLIENT.IPv6.src.subnet(32)" CLIENT.IPv6.dst Q.URL
add ns limitIdentifier ipv6_id -imeSlice 20000 -selectorName ipv6_sel
add lb vserver ipv6_vip HTTP 3ffe::209 80 -persistenceType NONE -cltTimeout 180
add responder action redirect_page redirect "\"http://redirectpage.com/\""
add responder policy ipv6_resp_pol "SYS.CHECK_LIMIT(\"ipv6_id\")" redirect_page
bind responder global ipv6_resp_pol 5 END -type DEFAULT
```

Sample Use Cases for Rate-Based Policies

September 21, 2020

The following scenarios describe two uses of rate-based policies in global server load balancing (GSLB):

- The first scenario describes the use of a rate-based policy that sends traffic to a new data center if the rate of DNS requests exceed 1000 per second.
- In the second scenario, if more than five DNS requests arrive for a local DNS (LDNS) client within a particular period, the additional requests are dropped.

Redirecting Traffic on the Basis of Traffic Rate

In this scenario, you configure a proximity-based load balancing method, and a rate-limiting policy that identifies DNS requests for a particular region. In the rate-limiting policy, you specify a threshold of 1000 DNS requests per second. A DNS policy applies the rate limiting policy to DNS requests for the region "Europe.GB.17.London.UK-East.ISP-UK." In the DNS policy, DNS requests that exceed the rate limiting threshold, starting with request 1001 and continuing to the end of the one-second interval, are to be forwarded to the IP addresses that are associated with the region "North America.US.TX.Dallas.US-East.ISP-US."

The following configuration demonstrates this scenario:

```
add stream selector DNSSelector1 client.udp.dns.domain

add ns limitIdentifier DNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName DNSSelector1

add dns policy DNSLimitPolicy1 "client.ip.src.matches_location(\"Europe.GB.17.London.*.*\") &&
sys.check_limit(\"DNSLimitIdentifier1\")" -preferredLocation "North America.US.TX.Dallas.*.*"

bind dns global DNSLimitPolicy1 5
<!--NeedCopy-->
```

Dropping DNS Requests on the Basis of Traffic Rate

In the following example of global server load balancing, you configure a rate limiting policy that permits a maximum of five DNS requests in a particular interval, per domain, to be directed to an LDNS client for resolution. Any requests that exceed this rate are dropped. This type of policy can help protect the NetScaler from resource exploitation. For example, in this scenario, if the time to live (TTL) for a connection is five seconds, this policy prevents the LDNS from requerying a domain. Instead, it uses data that is cached on the NetScaler.

```
add stream selector LDNSSelector1 client.udp.dns.domain client.ip.src

add ns limitIdentifier LDNSLimitIdentifier1 -threshold 5 -timeSlice 1000 -selectorName LDNSSelector1

add dns policy LDNSPolicy1 "client.udp.dns.domain.contains(\".\") && sys.check_limit(\"LDNSLimitIdentifier1\")" -drop YES

bind dns global LDNSPolicy1 6

show gslb vserver gvip

gvip - HTTP     State: UP
Last state change was at Mon Sep  8 11:50:48 2008 (+711 ms)
Time since last state change: 1 days, 02:55:08.830
Configured Method: STATICPROXIMITY
BackupMethod: ROUNDROBIN
No. of Bound Services :  3 (Total)       3 (Active)
Persistence: NONE       Persistence ID: 100
Disable Primary Vserver on Down: DISABLED       Site Persistence: NONE
Backup Session Timeout: 0
Empty Down Response: DISABLED
Multi IP Response: DISABLED Dynamic Weights: DISABLED
Cname Flag: DISABLED
Effective State Considered: NONE
1)      site11_svc(10.100.00.00: 80)- HTTP State: UP    Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: Europe.GB.17.London.UK-East.ISP-UK
2)      site12_svc(10.101.00.100: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.TX.Dallas.US-East.ISP-US
3)      site13_svc(10.102.00.200: 80)- HTTP State: UP   Weight: 1
Dynamic Weight: 0       Cumulative Weight: 1
Effective State: UP
Threshold : BELOW
Location: North America.US.NJ.Salem.US-Mid.ISP-US
```

```
1)      www.gslbindia.com        TTL: 5 secn
Cookie Timeout: 0 min   Site domain TTL: 3600 sec
Done
<!--NeedCopy-->
```

Rate Limiting for Traffic Domains

September 21, 2020

You can configure rate limiting for traffic domains. The following expression in the NetScaler expressions language for identifies traffic associated with traffic domains.

- client.traffic_domain.id

You can configure rate limiting for traffic associated with a particular traffic domain, a set of traffic domains, or all traffic domains.

For configuring rate limiting for traffic domains, you perform the following steps on a NetScaler appliance by using the GUI or the NetScaler command line:

1. Configure a stream selector that uses the client.traffic_domain.id expression for identifying the traffic, associated with traffic domains, to be rate limited.
2. Configure a rate limit identifier that specifies parameters such as maximum threshold for traffic to be rate limited. You also associate a stream selector to the rate limiter in this step.
3. Configure an action that you want to associate with the policy that uses the rate limit identifier.
4. Configure a policy that uses the sys.check_limit expression prefix to call the rate limit identifier, and associate the action with this policy.
5. Bind the policy globally.

Consider an example in which two traffic domains, with IDs 10 and 20, are configured on NetScaler NS1. On traffic domain 10, LB1-TD-1 is configured to load balance servers S1 and S2; LB2-TD1 is configured to load balance servers S3 and S4.

On traffic domain 20, LB1-TD-2 is configured to load balance servers S5 and S6; LB2-TD2 is configured to load balance servers S7 and S8.

The following table lists some examples of rate limiting policies for traffic domains in the example setup.

Purpose	CLI commands
Limit the number of requests to 10 per second for each of the traffic domains.	add stream selector tdratelimit-1 CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier limitidf-1 -threshold 10 -selectorName tdratelimit-1 -trapsInTimeSlice 0 add responder policy ratelimit-pol "sys.check_limit(\"limitidf-1\")" DROP bind responder global ratelimit-pol 1
Limit the number of requests to 5 per client per second for each of the traffic domains.	add stream selector tdandclientip CLIENT.IP.SRC,CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td_limitidf -threshold 5 -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy tdratelimit-pol "sys.check_limit(\"td_limitidf\")" DROP bind responder global tdratelimit-pol 2
Limit the number of requests sent for a particular traffic domain (for example traffic domain 10) to 30 requests every 3 seconds.	add stream selector tdratelimit CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td10_limitidf -threshold 30 -timeSlice 3000 -selectorName tdratelimit -trapsInTimeSlice 5 add responder policy td10ratelimit "client.traffic_domain.id==10 && sys.check_limit(\"td10_limitidf\")" DROP bind responder global td10ratelimit 3
Limit the number of connections to 5 per client per second for a particular traffic domain (for example traffic domain 20).	add stream selector tdandclientip CLIENT.IP.SRC CLIENT.TRAFFIC_DOMAIN.ID add ns limitIdentifier td20_limitidf -threshold 5 -mode CONNECTION -selectorName tdandclientip -trapsInTimeSlice 5 add responder policy td20_ratelimit "client.traffic_domain.id==20 && sys.check_limit(\"td20_limitidf\")" DROP bind responder global td20_ratelimit 4

Configure rate limit at packet level

September 21, 2020

Note: This feature was introduced in NetScaler release 11.1 build 51.21.

You can collect statistics at the packet level to determine bad/attack prone packets flowing through all the connections identified by the selector. At any point, if the percentage of bad or attack-prone packets exceed the configured threshold, a corrective action (RESET or DROP) is triggered as per the configuration. This functionality can be used to address DDoS attacks involving small TCP packets in which PUSH flag is enabled.

The following configuration demonstrates this functionality. This configuration tracks packet credits for all the TCP connections flowing through the system. This creates a session and associates on pcb/natpcb and the performs the per packet check.

Example:

```
add stream selector packetcreditrateselector client.ip.src client.tcp.srcport client.ip.dst client.tcp.dstport

add stream identifier packetcreditrateidentifier packetcreditrateselector  - interval 1

add responder policy packetcreditratesessionpolicy "ANALYTICS.STREAM(\"packetcreditrateidentifier\").COLLECT_STATS(\"PACKET_CREDITS\", <max_threshold_percentage>, ACTION)" NOOP

<max_threshold_percentage> is any value between 0-100.
<!--NeedCopy-->
```

ACTION can be either DROP/RESET

After the configuration is complete, we must bind this responder policy globally.

Example:

```
bind responder global packetcreditratesessionpolicy 101 END -type REQ_DEFAULT

bind responder global packetcreditratesessionpolicy 102 END -type NAT_REQ_DEFAULT
<!--NeedCopy-->
```

Responder

September 21, 2020

> **Warning**
>
> Responder and Rewrite filters are deprecated and as an alternative, Citrix recommends you to use filters using advanced policy infrastructure.

Today's complex Web configurations often require different responses to HTTP requests that appear, on the surface, to be similar. When users request a website's home page, you may want to provide a different home page depending on where each user is located, which browser the user is using, or which languages the browser accepts and the order of preference. You might want to break the connection immediately if the request is coming from an IP range that has been generating DDoS attacks or initiating hacking attempts.

Responder supports protocols such as TCP, DNS (UDP), and HTTP. With responder enabled on your appliance, server responses can be based on who sends the request, where it is sent from, and other criteria with security and system management implications. The feature is simple and quick to use. By avoiding the invocation of more complex features, it reduces CPU cycles and time spent in handling requests that do not require complex processing.

For handling sensitive data such as financial information, if you want to ensure that the client uses a secure connection to browse a site, you can redirect the request to secure connection by using `https://` instead of `http://`.

To use the Responder feature, do the following;

- Enable the Responder feature on the NetScaler.
- Configure responder actions. The action can be to generate a custom response, redirect a request to a different webpage, or reset a connection.
- Configure responder policies. The policy determines the requests (traffic) on which an action has to be taken.
- Bind each policy to a bind point put it into effect. A bind point refers to an entity at which NetScaler examines the traffic to see if it matches a policy. For example, a bind point can be a load balancing virtual server.

You can specify a default action for requests that do not match any policy, and you can bypass the safety check for actions that would otherwise generate error messages.

The Rewrite feature of NetScaler helps in rewriting some information in the requests or responses handled by NetScaler. The following section shows some differences between the two features.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a HTTP request to new websites or webpages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

Enabling the Responder Feature

September 21, 2020

To use the Responder feature, you must first enable it.

To enable the responder feature by using the NetScaler command line:

At the command prompt, type the following commands to enable the responder feature and verify the configuration:

- enable ns feature <feature>
- show ns feature

Example:

```
enable ns feature Responder
 Done
> show ns feature

        Feature                        Acronym             Status
        -------                        -------             ------
 1)     Web Logging                    WL                  ON
 2)     Surge Protection               SP                  ON
 .
 .
 .
 1)     Responder                      RESPONDER           ON
 2)     HTML Injection                 HTMLInjection       ON
 3)     NetScaler Push                 push                OFF
 Done
>
<!--NeedCopy-->
```

To enable the responder feature by using the GUI:

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Change advanced features.
3. In the Configure Advanced Features dialog box, select the Responder check box, and then click OK.
4. In the Enable/Disable Feature(s)? dialog box, click YES. A message appears in the status bar, stating that the feature has been enabled.

Configuring a Responder Action

September 21, 2020

After enabling the responder feature, you must configure one or more actions for handling requests. The responder supports the following types of actions:

- **Respond with**. Sends the response defined by the Target expression without forwarding the request to a web server. (The NetScaler appliance substitutes for and acts as a web server.) Use this type of action to manually define a simple HTML-based response. Normally the text for a Respond with action consists of a web server error code and brief HTML page.
- **Respond with SQL OK**. Sends the designated SQL OK response defined by the Target expression. Use this type of action to send an SQL OK response to an SQL query.
- **Respond with SQL Error**. Sends the designated SQL Error response defined by the Target expression. Use this type of action to send an SQL Error response to an SQL query.
- **Respond with HTML page**. Sends the designated HTML page as the response. You can choose from a drop-down list of HTML pages that were previously uploaded, or upload a new HTML page. Use this type of action to send an imported HTML page as the response.
- **Redirect**. Redirects the request to a different web page or web server. A Redirect action can redirect requests originally sent to a "dummy" web site that exists in DNS, but for which there is no actual web server, to an actual web site. It can also redirect search requests to an appropriate URL. Normally, the redirection target for a Redirect action consists of a complete URL.

To configure a responder action by using the NetScaler command line:

Displays the current settings for the specified responder action. If no action name is provided, displays a list of all responder actions currently configured on the NetScaler appliance, with abbreviated settings.

At the command prompt, type the following commands to configure a responder action and verify the configuration:

- add responder action <name> <type> <target> [-bypassSafetyCheck (YES | NO)]
- show responder action

Parameters:

- **Name**. Name of the responder action. Maximum Length: 127
- **type**. Type of responder action. It can be: (respondwith).
- **target**. An expression specifying what to respond with
- **htmlpage**. Option specifying to respondwith htmlpage
- **bypassSafetyCheck**. The safety check to allow unsafe expressions. **Note:** This attribute is deprecated.
- **hits**. The number of times the action has been taken.
- **referenceCount**. The number of references to the action.
- **undefHits**. The number of times the action resulted in UNDEF.
- **comment**. Any type of information about this responder action.
- **builtin**. Flag to determine whether responder action is built-in or not

Example:

```
To create a responder action that displays a “Not Found” error page
    for URLs that do not exist:

> add responder action act404Error respondWith '"HTTP/1.1 404 Not Found
    \r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web
    server."'
Done

> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does
    not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done

```

```
To create a responder action that displays a “Not Found” error page for URLs that do not exist:

add responder action act404Error respondWith '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
Done
> show responder action

1) Name: act404Error
Operation: respondwith
Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
BypassSafetyCheck : NO
Hits: 0
Undef Hits: 0
Action Reference Count: 0
Done
<!--NeedCopy-->
```

To modify an existing responder action by using the Citrix ADC command line:

At the command prompt, type the following command to modify an existing responder action and verify the configuration:

- `set responder action <name> -target <string> [-bypassSafetyCheck ( YES | NO )]`
- `show responder action`

Example:

```
set responder action act404Error  -target '"HTTP/1.1 404 Not Found\r\n\r\n"+ HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."'
 Done

> show responder action

1)       Name: act404Error
         Operation: respondwith
         Target: "HTTP/1.1 404 Not Found" + HTTP.REQ.URL.HTTP_URL_SAFE + " does not exist on the web server."
         BypassSafetyCheck : NO
         Hits: 0
         Undef Hits: 0
         Action Reference Count: 0
 Done

<!--NeedCopy-->
```

To remove a responder action by using the Citrix ADC command line:

At the command prompt, type the following command to remove a responder action and verify the configuration:

- `rm responder action <name>`
- `show responder action`

Example:

```
rm responder action act404Error
 Done

> show responder action
 Done

<!--NeedCopy-->
```

To configure a responder action by using the GUI:

1. Navigate to AppExpert > Responder > Actions.
2. In the details pane, do one of the following:
   - To create a new action, click Add.
   - To modify an existing action, select the action, and then click Open.
3. Click Create or OK, depending on whether you are creating a new action or modifying an existing action.
4. Click Close. A message appears in the status bar, stating that the feature has been enabled.
5. To delete a responder action, select the action, and then click Remove. A message appears in the status bar, stating that the feature has been disabled.

To add an expression by using the Add Expression dialog box

1. In the Create Responder Action or Configure Responder Action dialog box, click Add.

2. In the Add Expression dialog box, in the first list box choose the first term for your expression.

   - HTTP. The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

   - SYS. The protected web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

   - CLIENT. The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

- ANALYTICS. The analytics data associated with the request. Choose this if you want to examine request metadata.
- SIP. A SIP request. Choose this if you want to examine some aspect of a SIP request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Configuring the Global HTTP Action

You can configure the global HTTP action to invoke a responder action when an HTTP request times out. To configure this feature, you must first create the responder action that you want to invoke. Then, you configure the global HTTP timeout action to respond to a timeout with that responder action.

To configure the global HTTP action by using the NetScaler command line:

At the command prompt, type the following command:

- set ns httpProfile -reqTimeoutAction <`responder action name`>
- save ns config

For \<`responder action name`>, substitute the name of the responder action.

Configuring a Responder Policy

September 21, 2020

After you configure a responder action, you must next configure a responder policy to select the requests to which the NetScaler appliance should respond. A responder policy is based on a rule, which consists of one or more expressions. The rule is associated with an action, which is performed if a request matches the rule.

Note: For creating and managing responder policies, the GUI provides assistance that is not available at the NetScaler command prompt.

To configure a responder policy by using the NetScaler command line:

At the command prompt, type:

- add responder policy []-appFlowaction
- show responder policy

Example:

```
> add responder policy policyThree "CLIENT.IP.SRC.IN_SUBNET
   (222.222.0.0/16)" RESET
 Done
> show responder policy policyThree

        Name: policyThree
        Rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
        Responder Action: RESET
        UndefAction: Use Global
        Hits: 0
        Undef Hits: 0
 Done
<!--NeedCopy-->
```

To modify an existing responder policy by using the NetScaler command line:

At the command prompt, type:

- set responder policy [-rule] [-action] [-undefAction]
- show responder policy

To remove a responder policy by using the NetScaler command line:

At the command prompt, type:

- rm responder policy
- show responder policy

Example:

```
>rm responder policy pol404Error
 Done

> show responder policy
 Done
<!--NeedCopy-->
```

To configure a responder policy by using the GUI:

1. Navigate to AppExpert > Responder > Policies.
2. In the details pane, do one of the following:
    - To create a new policy, click Add.
    - To modify an existing policy, select the policy, and then click Open.
3. Click Create or OK, depending on whether you are creating a new policy or modifying an existing policy.
4. Click Close. A message appears in the status bar, stating that the feature has been configured.

Binding a Responder Policy

September 21, 2020

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to requests whose destination IP address is the VIP of that virtual server.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000. The responder feature implements only the first policy that a request matches, not any additional policies that it might also match, so policy priority is important for getting the results you intend.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add additional policies at any time without having to reassign the priority of an existing policy.

For additional information about binding policies on the NetScaler, see Policies and Expressions.

Note: Responder policies cannot be bound to TCP-based virtual servers.

To globally bind a responder policy by using the NetScaler command line:

At the command prompt, type the following command to globally bind a responder policy and verify the configuration:

- `bind responder global <policyName> <priority> [<gotoPriorityExpression [-type <type>] [-invoke (<labelType> <labelName>)]`
- `show responder global`

Example:

```
> bind responder global poliError 100
 Done
> show responder global
1)      Global bindpoint: REQ_DEFAULT
        Number of bound policies: 1

 Done
<!--NeedCopy-->
```

To bind responder policy to a specific virtual server by using the NetScaler command line:

At the command prompt, type:

- `bind lb vserver <name> -policyname <policy_name> -priority <priority>`
- `sh lb vserver <name>`

Example:

```
> bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
> show lb vserver
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
2)      vs-cont-sw (0.0.0.0:0) - TCP    Type: ADDRESS
        State: DOWN
        Last state change was at Wed Aug 19 10:03:46 2009 (+213 ms)
        Time since last state change: 2 days, 00:00:04.260
        Effective State: DOWN
        Client Idle Timeout: 9000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

To globally bind a responder policy by using the GUI:

1. Navigate to **AppExpert** > **Responder** > **Policies**.
2. On the **Responder Policies** page, select a responder policy, and then click **Policy Manager**.
3. In the **Responder Policy Manager** dialog box Bind Points menu, select Default Global.
4. Click **Insert Policy** to insert a new row and display a drop-down list of all unbound responder policies.
5. Click one of the policies on the list. That policy is inserted into the list of globally bound responder policies.
6. Click **Apply Changes**.
7. Click **Close**. A message appears in the status bar, stating that the configuration has been successfully completed.

To bind a responder policy to a specific virtual server by using the GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. On the **Load Balancing Virtual Servers** page, select the virtual server to which you want to bind the responder policy, and then click **Open**.
3. In the **Configure Virtual Server** (Load Balancing) dialog box, select the **Policies** tab, which displays a list of all policies configured on your NetScaler appliance.
4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click **OK**. A message appears in the status bar, stating that the configuration has been successfully completed.

Setting the default action for a responder policy

September 21, 2020

The NetScaler appliance generates an undefined event (UNDEF event) when a request does not match a responder policy, and then carries out the default action assigned to undefined events. By default, that action is to forward the request to the next feature such as load balancing, content filtering and so forth. This default behavior is normally what you want; it ensures that requests that do not require special handling by a specific responder action are sent to your Web servers and clients receive access to the content that they requested.

If the Web site(s) your NetScaler appliance protects receive a significant number of invalid or malicious requests, however, you may want to change the default action to either reset the client connection or drop the request. In this type of configuration, you would write one or more responder policies that would match any legitimate requests, and simply redirect those requests to their original destinations. Your NetScaler appliance would then block any other requests as specified by the default action you configured.

You can assign any one of the following actions to an undefined event:

- **NOOP**

  The NOOP action aborts responder processing but does not alter the packet flow. This means that the appliance continues to process requests that do not match any responder policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers and is the default setting.

- **RESET**

  If the undefined action is set to RESET, the appliance resets the client connection, informing the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

- **DROP**

  If the undefined action is set to DROP, the appliance silently drops the request without responding to the client in any way. This action is appropriate for requests that appear to be part of a DDoS attack or other sustained attack on your servers.

Note: UNDEF events are triggered only for client requests. No UNDEF events are triggered for responses.

To set the undefined action by using the NetScaler command line:

At the command prompt, type the following command to set the undefined action and verify the configuration:

```
set responder param          DROP                NOOP)
-undefAction (RESET
```

-
- show responder param

Example:

```
>set responder param -undefAction RESET
 Done
> show responder param
        Action Name: RESET
 Done
>
<!--NeedCopy-->
```

To set the undefined action by using the GUI:

1. Navigate to **AppExpert** > **Responder**, and then under **Settings**, click the **Change Responder Settings** link.
2. In the **Set Responder Params** dialog box, under **Global Undefined-Result Action**, select **NOOP**, **RESET**, or **DROP**.
3. Click **OK**. A message appears in the status bar, stating that the Responder Parameters have been configured.

Responder action and policy examples

September 21, 2020

Responder actions and policies are powerful and complex, but you can get started with relatively simple applications.

Example: Blocking access from specified IP addresses

The following procedures block access to your protected Web site(s) by clients originating from the CIDR 222.222.0.0/16. The responder sends an error message stating that the client is not authorized to access the URL requested.

To block access by using the NetScaler command line:

At the command prompt, type the following commands to block access:

- add responder action act_unauthorized respond with “HTTP/1.1 403 Forbidden\r\n\r\n” + “Client: “ + CLIENT.IP.SRC + “ is not authorized to access URL:” + “HTTP.REQ.URL.HTTP_URL_SAFE”’
- add responder policy pol_un “CLIENT.IP.SRC.IN_SUBNET (222.222.0.0/16)” act_unauthorized
- bind responder global pol_un 10

To block access by using the GUI:

1. In the navigation pane, expand **Responder**, and then click **Actions**.
2. In the details pane, click **Add**.

3. In the **Create Responder Action** dialog box, do the following:
    a) In the Name text box, type act_unauthorized.
    b) Under Type, select Respond with.
    c) In the Target text area, type the following string: "HTTP/1.1 200 OK\r\n\r\n" + "Client: " + CLIENT.IP.SRC + " is not authorized to access URL:" + HTTP.REQ.URL.HTTP_URL_SAFE
    d) Click **Create**, and then click **Close**.
    The responder action you configured, named act_unauthorized, now appears in the Responder Actions page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the **Create Responder Policy** dialog box, do the following:
    a) In the Name text box, type pol_unauthorized.
    b) Under Action, select act_unauthorized.
    c) In the Expression window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
    d) Click **Create**, then click **Close**.
    The responder policy you configured, named pol_unauthorized, now appears in the **Responder Policies** page.
7. Globally bind your new policy, pol_unauthorized, as described in Binding a Responder Policy.

Example: Redirecting a client to a new URL

The following procedures redirect clients who access your protected Web site(s) from within the CIDR 222.222.0.0/16 to a specified URL.

To redirect clients by using the NetScaler command line:

At the command prompt, type the following commands to redirect clients and verify the configuration:

- `add responder action act_redirect redirect '"<http://www.example.com/404.html>"'`
- `show responder action act_redirect`
- `add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)" act_redirect`
- `show responder policy pol_redirect`
- `bind responder global pol_redirect 10`

Example:

```
> add responder action act_redirect redirect '" http ://www.example.com/404.html "'
Done

```

```
> add responder policy pol_redirect "CLIENT.IP.SRC.IN_SUBNET
   (222.222.0.0/16)" act_redirect
Done
<!--NeedCopy-->
```

To redirect clients by using the GUI:

1. Navigate to **AppExpert** > **Responder** > **Actions**.
2. In the details pane, click **Add**.
3. In the **Create Responder Action** dialog box, do the following:
   a) In the Name text box, type act_redirect.
   b) Under Type, select **Redirect**.
   c) In the Target text area, type the following string: `"<http://www.example.com/404.html>"`
   d) Click , then **Create** click **Close**.
      The responder action you configured, named act_redirect, now appears in the Responder Actions page.
4. In the navigation pane, click **Policies**.
5. In the details pane, click **Add**.
6. In the Create Responder Policy dialog box, do the following:
   a) In the **Name** text box, type pol_redirect.
   b) Under **Action**, select act_redirect.
   c) In the Expression window, type the following rule: CLIENT.IP.SRC.IN_SUBNET(222.222.0.0/16)
   d) Click **Create**, then click **Close**.
      The responder policy you configured, named pol_redirect, now appears in the **Responder Policies** page.
7. Globally bind your new policy, pol_redirect, as described in Binding a Responder Policy.

Diameter Support for Responder

September 21, 2020

The Responder feature now supports the Diameter protocol. You can configure Responder to respond to Diameter requests as it does HTTP and TCP requests. For example, you could configure Responder to respond to requests from a specific Diameter origin with a redirect to a web page enhanced for mobile devices. A number of NetScaler expressions have been added that support examination of the Diameter header and the attribute-value pairs (AVPs). These expressions support lookup of specific AVPs by index, ID or name, examine the information in each AVP, and send an appropriate response.

To configure Responder to respond to a Diameter request:

At the command prompt, type the following commands:

- add responder action <actname> RESPONDWITH “DIAMETER.NEW_REDIRECT(”aaa://host.example.com”)”

  For <actname>, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For aaa://host.example.com, substitute the URL of the diameter host to which you want to redirect connections.

- add responder policy <polname> “diameter.req.avp(264).value.eq(”host1.example.net”)” <actname>

  For <polname>, substitute a name for your new policy. As with <actname>, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host1.example.net, substitute the name of the originating host of the requests that you want to redirect. For <actname>, substitute the name of the action that you just created.

- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST

  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. For <polname>, substitute the name of the policy you just created. For <priority>, substitute a priority for the policy.

Example:

To create a Responder action and policy to respond to Diameter requests that originate from “host1.example.net” with a redirect to “host.example.com”, you could add the following action and policy, and bind the policy as shown.

```
> add responder action act_resp-dm-redirect RESPONDWITH "DIAMETER.NEW_REDIRECT(\"aaa://host.example.com\")"
Done

> add responder pol_resp-dm-redirect "diameter.req.avp(264).value.eq(\"host1.example.net\")" act_resp-dm-redirect
Done

> bind lb vserver vs1 -policyName pol_resp-dm-redirect -priority 10 -type REQUEST
Done
<!--NeedCopy-->
```

RADIUS Support for Responder

September 21, 2020

The NetScaler expressions language contains expressions that can extract information from and manipulate RADIUS requests. These expressions enable you to use the Responder feature to respond to RADIUS requests. Your responder policies and actions can use any expression that is appropriate or relevant to a RADIUS request. The available expressions enable you to identify the RADIUS message type, extract any attribute-value pair (AVP) from the connection, and send different responses on the basis of that information. You can also create policy labels that invoke all responder policies for RADIUS connections.

You can use RADIUS expressions to construct simple responses that do not require communication with the RADIUS server to which the request was sent. When a responder policy matches a connection, the NetScaler constructs and sends the appropriate RADIUS response without contacting the RADIUS authentication server. For example, if the source IP address of a RADIUS request is from a subnet that is specified in the responder policy, the NetScaler can reply to that request with an access-reject message, or can simply drop the request.

You can also create policy labels to route specific types of RADIUS requests through a series of policies that are appropriate to those requests.

Note: The current RADIUS expressions do not work with RADIUS IPv6 attributes.

The NetScaler documentation for expressions that support RADIUS assumes familiarity with the basic structure and purpose of RADIUS communications. If you need more information about RADIUS, see your RADIUS server documentation or search online for an introduction to the RADIUS protocol.

Configuring Responder Policies for RADIUS

The following procedure uses the NetScaler command line to configure a responder action and policy, and bind the policy to a RADIUS-specific global bind point.

To configure a Responder action and policy, and bind the policy:

At the command prompt, type the following commands:

- add responder action <actName> <actType>
- add responder policy <polName> <rule> <actName>
- bind responder policy <polName> <priority> <nextExpr> -type <bindPoint> where <bindPoint> represents one of the RADIUS-specific global bind points.

RADIUS Expressions for Responder

In a responder configuration, you can use the following NetScaler expressions to refer to various portions of a RADIUS request.

Identifying the Type of Connection:

- RADIUS.IS_CLIENT

  Returns TRUE if the connection is a RADIUS client (request) message.

- RADIUS.IS_SERVER

  Returns TRUE if the connection is a RADIUS server (response) message.

Request Expressions:

- RADIUS.REQ.CODE

  Returns the number that corresponds to the RADIUS request type. A derivative of the num_at class. For example, a RADIUS access request would return 1 (one). A RADIUS accounting request would return 4.

- RADIUS.REQ.LENGTH

  Returns the length of the RADIUS request, including the header. A derivative of the num_at class.

- RADIUS.REQ.IDENTIFIER

  Returns the RADIUS request identifier, a number assigned to each request that allows the request to be matched to the corresponding response. A derivative of the num_at class.

- RADIUS.REQ.AVP(<AVP Code No>).VALUE

  Returns the value of first occurrence of this AVP as a string of type text_t.

- RADIUS.REQ.AVP(<AVP code no>).INSTANCE(instance number)

  Returns the specified instance of the AVP as a string of type RAVP_t. A specific RADIUS AVP can occur multiple times in a RADIUS message. INSTANCE (0) returns the first instance, INSTANCE (1) returns second instance, and so on, up to sixteen instances.

- RADIUS.REQ.AVP(<AVP code no>).VALUE(instance number)

  Returns the value of specified instance of the AVP as a string of type text_t.

- RADIUS.REQ.AVP(<AVP code no>).COUNT

  Returns the number of instances of a specific AVP in a RADIUS connection, as an integer.

- RADIUS.REQ.AVP(<AVP code no>).EXISTS

  Returns TRUE if the specified type of AVP exists in the message, or FALSE if it does not.

Response Expressions:

RADIUS response expressions are identical to RADIUS request expressions, except that RES replaces REQ.

Typecasts of AVP Values:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

Example:

The ADC supports expressions to typecast RADIUS AVP values to the text, integer, unsigned integer, long, unsigned long, ipv4 address, ipv6 address, ipv6 prefix and time data types. The syntax is the same as for other NetScaler typecast expressions.

```
RADIUS.REQ.AVP(8).VALUE(0).typecast_ip_address_at
<!--NeedCopy-->
```

AVP Type Expressions:

The NetScaler supports expressions to extract RADIUS AVP values by using the assigned integer codes described in RFC2865 and RFC2866. You can also use text aliases to accomplish the same task. Some examples follow.

- RADIUS.REQ.AVP (1).VALUE or RADIUS.REQ.USERNAME.value

  Extracts the RADIUS user-name value.

- RADIUS.REQ.AVP (4). VALUE or RADIUS.REQ. ACCT_SESSION_ID.value

  Extracts the Acct-Session-ID AVP (code 44) from the message.

- RADIUS.REQ.AVP (26). VALUE or RADIUS.REQ.VENDOR_SPECIFIC.VALUE

  Extracts the vendor-specific value.

The values of most commonly-used RADIUS AVPs can be extracted in the same manner.

RADIUS Bind Points:

Four global bind points are available for policies that contain RADIUS expressions.

- RADIUS_REQ_OVERRIDE

  Priority/override request policy queue.

- RADIUS_REQ_DEFAULT

  Standard request policy queue.

- RADIUS_RES_OVERRIDE

  Priority/override response policy queue.

- RADIUS_RES_DEFAULT

  Standard response policy queue.

RADIUS Responder-Specific Expressions:

- RADIUS_RESPONDWITH

  Respond with the specified RADIUS response. The response is created with NetScaler expressions, both RADIUS expressions and any others that are applicable.

- RADIUS.NEW_ANSWER

  Sends a new RADIUS answer to the user.

- RADIUS.NEW_ACCESSREJECT

  Rejects the RADIUS request.

- RADIUS.NEW_AVP

  Adds the specified new AVP to the response.

Use Cases

Following are use cases for RADIUS with responder.

Blocking RADIUS Requests from a Specific Network

To configure the responder feature to block authentication requests from a specific network, begin by creating a responder action that rejects requests. Use the action in a policy that selects requests from the networks that you want to block. Bind the responder policy to a RADIUS-specific global bind point, specifying:

- The priority
- END as the nextExpr value, to ensure that policy evaluation stops when this policy is matched
- RADIUS_REQ_OVERRIDE as the queue to which you assign the policy, so that it is evaluated before policies assigned to the default queue

To configure Responder to block logons from a specific network**

- add responder action <actName> <actType>
- add responder policy <polName> <rule> <actName>

- bind responder global <polName> <priority> <nextExpr> -type <bindPoint>

Example:

```
> add responder action rspActRadiusReject respondwith radius.
    new_accessreject
Done

> add responder policy rspPolRadiusReject client.ip.src.in_subnet
    (10.224.85.0/24) rspActRadiusReject
Done

> bind responder global rspPolRadiusReject 1 END -type
    RADIUS_REQ_OVERRIDE
<!--NeedCopy-->
```

DNS Support for the Responder Feature

September 21, 2020

You can configure the responder feature to respond to DNS requests as it does to HTTP and TCP requests. For example, you could configure it to send DNS responses over UDP and ensure that the DNS requests from the client are sent over TCP. A number of NetScaler expressions support examination of the DNS header in the request. These expressions examine specific header fields and send an appropriate response.

- **DNS Expressions**. In a responder configuration, you can use the following NetScaler expressions to refer to various portions of a DNS request:

Expressions	Descriptions
DNS.NEW_RESPONSE	Creates a new empty DNS response based on the request.
DNS.NEW_RESPONSE <AA, TC, rcode>	Creates a new DNS response based on the specified parameters.

- **DNS Bind Points**. The following global bind points are available for policies that contain DNS expressions.

Bind Points	Descriptions
DNS_REQ_OVERRIDE	Priority/override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.

In addition to the default bind points, you can create policy labels of type DNS and bind DNS policies to them.

Configuring Responder Policies for DNS

The following procedure uses the NetScaler command line to configure a responder action and policy and bind the policy to a responder-specific global bind point.

To configure Responder to respond to a DNS request:

At the command prompt, type the following commands:

1. add responder action <actName> <actType>

   For <actname>, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actType>, substitute a responder action type, *respondWith*.

2. add responder policy <polName> <rule> <actName>

   For <polname>, substitute a name for your new policy. For <actname>, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actname>, substitute the name of the action that you just created.

3. bind responder policy <polName> <priority> <nextExpr> -type <bindPoint>

   For <bindPoint>, specify one of the responder-specific global bind points. For <polName>, substitute the name of the policy that you just created. For <priority>, specify the priority of the policy.

Sample configuration - Enforce all DNS request over TCP:

To enforce all the DNS requests over TCP, create a responder action that will set the TC bit and rcode as NOERROR.

```
> add responder action resp_act_set_tc_bit respondwith DNS.NEW_RESPONSE
    (true, true, NOERROR)
Done

> add responder policy enforce_tcp  dns.REQ.TRANSPORT.EQ(udp)
    resp_act_set_tc_bit
```

```
Done

>bind lb vserver dns_udp - policyName  enforce_tcp   -type request -
    priority 100
Done
<!--NeedCopy-->
```

Troubleshooting

September 21, 2020

If the responder feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting

For best results, use the following resources to troubleshoot an integrated cache issue on a NetScaler appliance:

- The ns.conf file
- The relevant trace files from the client and the NetScaler appliance

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the NetScaler trace files

Troubleshooting Responder Issues

- **Issue**

  The Responder feature is configured, but the responder action is not working.

  Resolution

  - Verify that the feature is enabled.
  - Check the hit counters of any of the policies to see if the counters are getting incremented.
  - Verify that the policies and actions are configured correctly.
  - Verify that the actions and policies are bound appropriately.
  - Record the packet traces on the client and the NetScaler appliance, and analyze the them to get some pointer to the issue.
  - Record the iehttpHeaters packet traces on the client and verify the HTTP requests and responses to get some pointer to the issue.

- **Issue**

  You need to create a maintenance page.

  Resolution

  1. Configure the services and virtual Server.
  2. Configure a backup virtual server with a service bound to it. This ensures that the status of the Web site is always displayed as UP.
  3. Configure the primary virtual server to use the backup virtual server as a backup.
  4. Create a responder action with an appropriate target. Following is an example for your reference:

     ```
     add responder action sorry_page respondwith q{“HTTP/1.0 200 OK” +”\r\n\r\n” + “<html><body>Sorry, this page is not available</body></html>” + “\r\n”} .
     ```

  5. Create a responder policy and bind the action to it.
  6. Bind the responder policy to the backup virtual Server.

Rewrite

September 21, 2020

May 24, 2018

> **Warning**
>
> Responder and Rewrite filters are deprecated and as an alternative, Citrix recommends you to use filters using Advanced policy infrastructure.

Rewrite refers to the rewriting of some information in the requests or responses handled by the NetScaler appliance. Rewriting can help in providing access to the requested content without exposing unnecessary details about the Web site’s actual configuration. A few situations in which the rewrite feature is useful are described below:

- <To improve security, the NetScaler can rewrite all the `http://links` to `https://` in the response body.
- <In the SSL offload deployment, the insecure links in the response have to be converted into secure links. Using the rewrite option, you can rewrite all the `http://links` to `https://` for making sure that the outgoing responses from NetScaler to the client have the secured links.
- <If a Web site has to show an error page, you can show a custom error page instead of the default 404 Error page. For example, if you show the home page or site map of the Web site instead of an error page, the visitor remains on the site instead of moving away from the Web site.

- <If you want to launch a new Web site, but use the old URL, you can use the Rewrite option.
- <When a topic in a site has a complicated URL, you can rewrite it with a simple, easy-to-remember URL (also referred to as 'cool URL').
- <You can append the default page name to the URL of a Web site. For example, if the default page of a company's Web site is `http://www.abc.com/index.php`, when the user types 'abc.com' in the address bar of the browser, you can rewrite the URL to 'abc.com/index.php'.

When you enable the rewrite feature, NetScaler can modify the headers and body of HTTP requests and responses.

To rewrite HTTP requests and responses, you can use protocol-aware NetScaler policy expressions in the rewrite policies you configure. The virtual servers that manage the HTTP requests and responses must be of type
HTTP or
SSL. In HTTP traffic, you can take the following actions:

- <Modify the URL of a request
- <Add, modify or delete headers
- <Add, replace, or delete any specific string within the body or headers.

To rewrite TCP payloads, consider the payload as a raw stream of bytes. Each of the virtual servers that managing the TCP connections must be of type TCP or SSL_TCP. The term TCP rewrite is used to refer to the rewrite of TCP payloads that are not HTTP data. In TCP traffic, you can add, modify, or delete any part of the TCP payload.

For examples to use the rewrite feature, see Rewrite Action and Policy Examples.

Comparison between Rewrite and Responder options

The main difference between the rewrite feature and the responder feature is as follows:

Responder cannot be used for response or server-based expressions. Responder can be used only for the following scenarios depending on client parameters:

- Redirecting a http request to new Web sites or Web pages
- Responding with some custom response
- Dropping or resetting a connection at request level

In case of a responder policy, the NetScaler examines the request from the client, takes action according to the applicable policies, sends the response to the client, and closes the connection with the client.

In case of a rewrite policy, the NetScaler examines the request from the client or response from the server, takes action according to the applicable policies, and forwards the traffic to the client or the server.

In general, it is recommended to use responder if you want the NetScaler to reset or drop a connection based on a client or request-based parameter. Use responder to redirect traffic, or respond with custom messages. Use rewrite for manipulating data on HTTP requests and responses.

How rewrite works

September 21, 2020

A rewrite policy consists of a rule and action. The rule determines the traffic on which rewrite is applied and the action determines the action to be taken by the NetScaler. You can define multiple rewrite policies. For each policy, specify the bind point and priority.

A bind point refers to a point in the traffic flow at which the NetScaler examines the traffic to verify whether any rewrite policy can be applied to it. You can bind a policy to a specific load balancing or content switching virtual server, or make the policy global if you want the policy to be applied to the entire traffic handled by the NetScaler. These policies are referred to as global policies.

In addition to the user-defined policies, the NetScaler has some default policies. You cannot modify or delete a default policy.

For evaluating the policies, NetScaler follows the order mentioned below:

- Global policies
- Policies bound to specific virtual servers
- Default policies

Note: NetScaler can apply a rewrite policy only when it is bound to a point.

NetScaler implements the rewrite feature in the following steps:

- The NetScaler appliance checks for global policies and then checks for policies at individual bind points.
- If multiple policies are bound to a bind point, the NetScaler evaluates the policies in the order of their priority. The policy with the highest priority is evaluated first. After evaluating each policy, if the policy is evaluated to TRUE (the traffic matches the rule), it adds the action associated with the policy to a list of actions to be performed. A match occurs when the characteristics specified in the policy rule match the characteristics of the request or response being evaluated.
- <For any policy, in addition to the action, you can specify the policy that should be evaluated after the current policy is evaluated. This policy is referred to as the ‘Go to Expression’. For any policy, if a Go to Expression (gotoPriorityExpr) is specified, the NetScaler evaluates the Go to Expression policy; it ignores policy with the next highest priority.

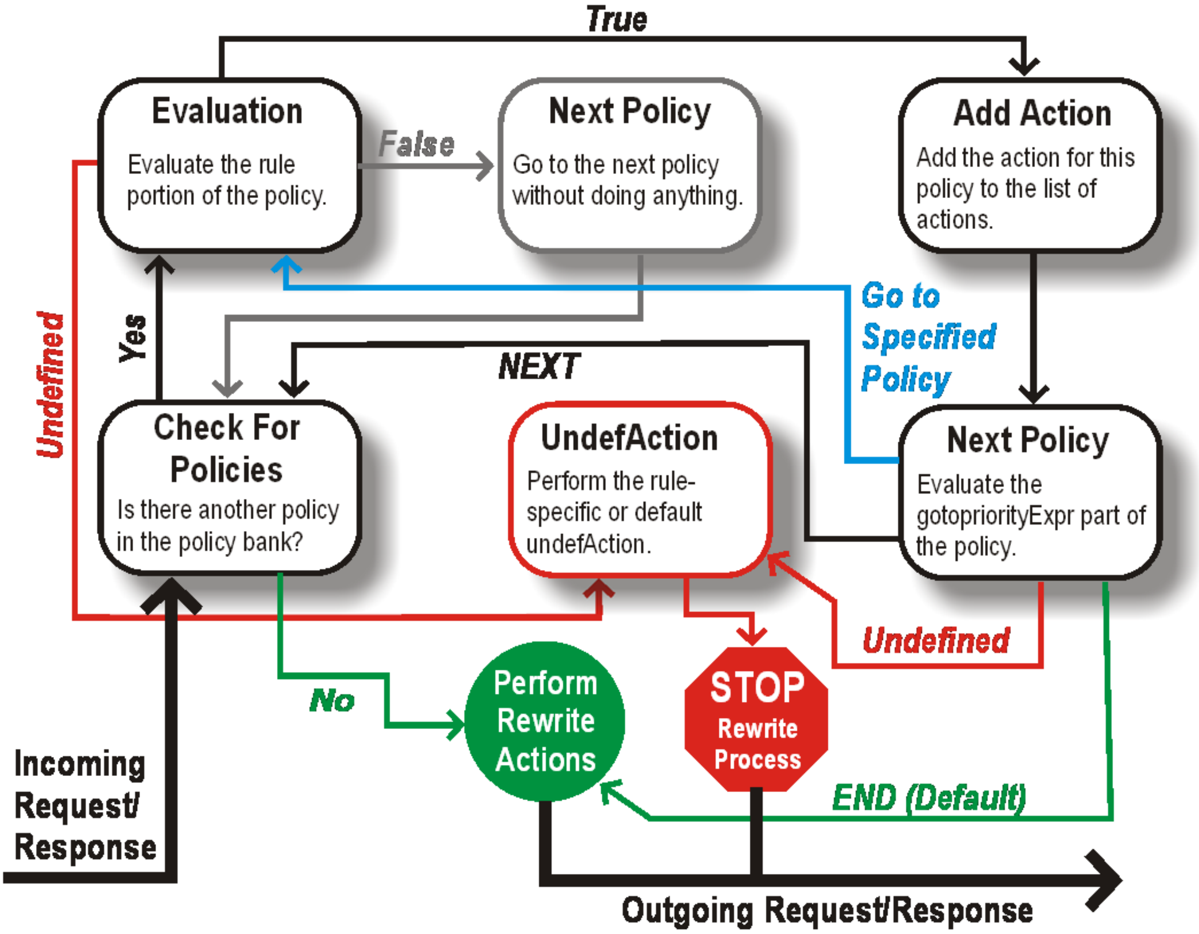
You can specify the priority of the policy to indicate the Go to Expression policy; you cannot use the name of the policy. If you want the NetScaler to stop evaluating other policies after evaluating a particular policy, you can set the Go to Expression to 'END'.

- After all the policies are evaluated or when a policy has the Go to Expression set as END, the NetScaler starts performing the actions according to the list of actions.

For more information about configuring rewrite policies, see "Configuring a Rewrite Policy" and about binding rewrite policies, see "Binding a Rewrite Policy."

The following figure illustrates how NetScaler processes a request or response when the rewrite feature is used.

Figure 1. The Rewrite Process


True
Evaluation
Evaluate the rule portion of the policy.
False
Next Policy
Go to the next policy without doing anything.
Add Action
Add the action for this policy to the list of actions.
Go to Specified Policy
Undefined
Yes
NEXT
Check For Policies
Is there another policy in the policy bank?
UndefAction
Perform the rule-specific or default undefAction.
Next Policy
Evaluate the gotopriorityExpr part of the policy.
Undefined
No
Perform Rewrite Actions
STOP Rewrite Process
END (Default)
Incoming Request/ Response
Outgoing Request/Response

Policy Evaluation

The policy with the highest priority is evaluated first. NetScaler does not stop the evaluation of rewrite policies when it finds a match; it evaluates all the rewrite policies configured on the NetScaler.

- If a policy evaluates to TRUE, the NetScaler follows the procedure below:

  - If the policy has the Go to Expression set to END, the NetScaler stops evaluating all the other policies and starts performing the rewrite.
  - The gotoPriorityExpression can be set to 'NEXT', 'END', some integer or 'INVOCATION_LIST'. The value determines the policy with the next priority. The following table shows the action taken by NetScaler for each value of the expression.
    | | |
    |—————————|—————————————————————————|
    | Value of the expression | Action |
    | NEXT | Policy with the next priority gets evaluated. |
    | END | Evaluation of policies stops. |
    | <an integer> | Policy with specified priority gets evaluated. |
    | INVOCATION_LIST | Goto NEXT or END is applied based on the result of the invocation list. |
- If a policy evaluates to FALSE, the NetScaler continues the evaluation in the order of priority.
- If a policy evaluates to UNDEFINED (cannot be evaluated on the received traffic due to an error), the NetScaler performs the action assigned to the UNDEFINED condition (referred to as undefAction) and stops further evaluation of polices.

The NetScaler starts the actual rewriting only after the evaluation is complete. It refers to the list of actions identified by policies that are evaluated to TRUE, and starts the rewriting. After implementing all the actions in the list, the NetScaler forwards the traffic as required.

Note: Ensure that the policies do not specify conflicting or overlapping actions on the same part of the HTTP header or body, or TCP payload. When such a conflict occurs, the NetScaler encounters an undefined situation and aborts the rewrite.

Rewrite Actions

On the NetScaler appliance, specify the actions to be taken such as adding, replacing, or deleting text within the body, or adding, modifying or deleting headers, or any changes in the TCP payload as rewrite actions. For more information about rewrite actions, see Configuring a Rewrite Action.

The following table describes the steps the NetScaler can take when a policy evaluates to TRUE.

Action	Result
Insert	The rewrite action specified for the policy is carried out.
NOREWRITE	The request or response is not rewritten. NetScaler forwards the traffic without rewriting any part of the message.

RESET	The connection is aborted at the TCP level.
DROP	The message is dropped.

Note: For any policy, you can configure the undefaction (action to be taken when the policy evaluates to UNDEFINED) as NOREWRITE, RESET, or DROP.

To use the
Rewrite feature, take the following steps:

- Enable the feature on the NetScaler.
- Define rewrite actions.
- Define rewrite policies.
- Bind the policies to a bind point to bring a policy into effect.

Enabling the rewrite feature

September 21, 2020

Enable the rewrite feature on the NetScaler appliance if you want to rewrite the HTTP or TCP requests or responses. If the feature is enabled, NetScaler takes rewrite action according to the specified policies. For more information, see
How rewrite works.

To enable the rewrite feature by using the command line interface

At the command prompt, type the following commands to enable the rewrite feature and verify the configuration:

- enable ns feature REWRITE
- show ns feature

Example:

```
> enable ns feature REWRITE
 Done
> show ns feature

        Feature                                Acronym                Status
        -------                                -------                ------
 1)     Web Logging                    WL                   OFF
 2)     Surge Protection               SP                   ON
 .
 .
 .
 1)     Rewrite                        REWRITE              ON
 .
 .
 1)     NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

To enable the rewrite feature by using the GUI

1. <In the navigation pane, click System, and then click Settings.
2. <In the details pane, under Modes and Features, click Configure basic features.
3. <In the Configure Basic Features dialog box, select the Rewrite check box, and then click OK.
4. <In the Enable/Disable Feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature was enabled.

Configuring a Rewrite Action

September 21, 2020

> **Warning**
>
> The Pattern function in a rewrite action is deprecated from NetScaler 12.0 build 56.20 onwards and as an alternative, Citrix recommends you to use the Search rewrite action parameter.

After enabling the rewrite feature, you need to configure one or more actions unless a built-in rewrite action is sufficient. All of the built-in actions have names beginning with the string ns_cvpn, followed by a string of letters and underscore characters. Built-in actions perform useful and complex tasks such as decoding parts of a clientless VPN request or response or modifying JavaScript or XML data. The built-in actions can be viewed, enabled, and disabled, but cannot be modified or deleted.

Target expressions in actions for TCP rewrite must begin with one of the following expression prefixes:

- <**CLIENT.TCP.PAYLOAD.** For rewriting TCP payloads in client requests. For example, CLIENT.TCP.PAYLOAD(10000).AFTER_STR("string1").
- <**SERVER.TCP.PAYLOAD.** For rewriting TCP payloads in server responses. For example, SERVER.TCP.PAYLOAD(1000).B64DECODE.BETWEEN("string1","string2").

You can use all types of existing string manipulation functions with these prefixes to identify the strings that you want to rewrite. To configure a rewrite action, you assign it a name, specify an action type, and add one or more arguments specifying additional data. The following table describes the action types and the arguments you use with them.

Note: Action types that can be used only for HTTP rewrite are identified in the **Rewrite Action Type** column.

See the Rewrite Action Types and their Arguments table.

To create a new rewrite action by using the command line interface

At the command prompt, type the following commands to create a new rewrite action and verify the configuration:

- `add rewrite action <name> <type> <target> [<stringBuilderExpr>] [(-pattern <expression> | -patset <string>)] [-bypassSafetyCheck (YES|NO)]`
- `show rewrite action <name>`

Example 1. Inserting an HTTP Header With the Client IP:

```
> add rewrite action insertact INSERT_HTTP_HEADER "client-IP" CLIENT.IP.SRC
Done

> show rewrite action insertact

        Name: insertact
        Operation: insert_http_header   Target:Client-IP
        Value:CLIENT.IP.SRC
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
<!--NeedCopy-->
```

Example 2. Replacing Strings in a TCP Payload (TCP Rewrite):

```
> add rewrite action client_tcp_payload_replace_all REPLACE_ALL
 'client.tcp.payload(1000)' '"new-string"' -search text("old-string")
 Done
> show rewrite action client_tcp_payload_replace_all

        Name: client_tcp_payload_replace_all
        Operation: replace_all
        Target:client.tcp.payload(1000)
        Value:"new-string"
        Search: text("old-string")
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
>
<!--NeedCopy-->
```

To modify an existing rewrite action by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite action and verify the configuration:

<set rewrite action <name> [-target <string>] [-stringBuilderExpr <string>] [(-pattern <expression>	-patset <string>)] [-bypassSafetyCheck (YES	NO)]

-
- <show rewrite action <name>

Example:

```
> set rewrite action insertact -target "Client-IP"
 Done
> show rewrite action insertact

        Name: insertact
        Operation: insert_http_header   Target:Client-IP
        Value:CLIENT.IP.SRC
        BypassSafetyCheck : NO
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 0
 Done
<!--NeedCopy-->
```

To remove a rewrite action by using the command line interface

At the command prompt, type the following commands to remove a rewrite action :

rm rewrite action <name>

Example:

```
> rm rewrite action insertact
Done
<!--NeedCopy-->
```

To configure a rewrite action by using the GUI

1. Navigate to AppExpert > Rewrite > Actions.
2. In the details pane, do one of the following:
    - To create a new action, click Add.
    - To modify an existing action, select the action, and then click Open.
3. Click Create or OK. A message appears in the status bar, stating that the Action has been configured successfully.
4. Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
5. Click Close.

To add an expression by using the Add Expression dialog box

1. In the Create Rewrite Action or Configure Rewrite Action dialog box, under the text area for the type argument you want to enter, click Add.

2. In the Add Expression dialog box, in the first list box choose the first term for your expression.

    - HTTP

      The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

    - SYS

      The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

    - CLIENT

      The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

   When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.
   For more information about the PI expressions language and creating expressions for responder policies, see "Policies and Expressions."

   If you want to test the effect of a rewrite action when used on sample HTTP data, you can use the Rewrite Expression Evaluator.

   Note: The Rewrite Expression Evaluator is only available in the GUI. There is no NetScaler command line version.

To evaluate a rewrite action by using the Rewrite Action Evaluator dialog box

1. In the Rewrite Actions details pane, select the rewrite action that you want to evaluate, and then click Evaluate.
2. In the Rewrite Expression Evaluator dialog box, specify values for the following parameters. (An asterisk indicates a required parameter.)
   - Rewrite Action*—If the rewrite action you want to evaluate is not already selected, select it from the drop-down list. After you select a Rewrite action, the Details section displays the details of the selected Rewrite action.
   - New*—Select New to open the Create Rewrite Action dialog box and create a new rewrite action.
   - Modify*—Select Modify to open the Configure Rewrite Action dialog box and modify the selected rewrite action.
   - Flow Type*—Specifies whether to test the selected rewrite action with HTTP Request data or HTTP Response data. The default is Request. If you want to test with Response data, select Response.
   - HTTP Request/Response Data*—Provides a space for you to provide the HTTP data that the Rewrite Action Evaluator will use for testing. You can paste the data directly into the window, or click Sample to insert some sample HTTP headers.
   - Show end-of-line—Specifies whether to show UNIX-style end-of-line characters (\n) at the end of each line of sample HTTP data.
   - Sample—Inserts sample HTTP data into the HTTP Request/Response Data window. You can choose either GET or POST data.

- Browse—Opens a local browse window so that you can choose a file containing sample HTTP data from a local or network location.
- Clear—Clears the current sample HTTP data from the HTTP Request/Response Data window.

3. Click Evaluate. The Rewrite Action Evaluator evaluates the effect of the Rewrite action on the sample data that you chose, and displays the results as modified by the selected Rewrite action in the Results window. Additions and deletions are highlighted as indicated in the legend in the lower left-hand corner of the dialog box.
4. Continue evaluating Rewrite actions until you have determined that all of your actions have the effect that you wanted.
    - You can modify the selected rewrite action and test the modified version by clicking Modify to open the Configure Rewrite Action dialog box, making and saving your changes, and then clicking Evaluate again.
    - You can evaluate a different rewrite action using the same request or response data by selecting it from the Rewrite Action drop-down list, and then clicking Evaluate again.
5. Click Close to close the Rewrite Expression Evaluator and return to the Rewrite Actions pane.

    To delete a rewrite action, select the rewrite action you want to delete, then click Remove and, when prompted, confirm your choice by clicking OK.

Configuring a Rewrite Policy

September 21, 2020

After you create any needed rewrite action(s), you must create at least one rewrite policy to select the requests that you want the NetScaler appliance to rewrite.

A rewrite policy consists of a rule, which itself consists of one or more expressions, and an associated action that is performed if a request or response matches the rule. Policy rules for evaluating HTTP requests and responses can be based on almost any part of a request or response.

Even though you cannot use TCP rewrite actions to rewrite data other than the TCP payload, you can base the policy rules for TCP rewrite policies on the information in the transport layer and the layers below the transport layer.

If a configured rule matches a request or response, the corresponding policy is triggered and the action associated with it is carried out.

Note: You can use either the command line interface or the GUI to create and configure rewrite policies. Users who are not thoroughly familiar with the command line interface and the NetScaler Policy expression language will usually find using the GUI much easier.

To add a new rewrite policy by using the command line interface

At the command prompt, type the following commands to add a new rewrite policy and verify the configuration:

- <add rewrite policy <name> <expression> <action> [<undefaction>]
- <show rewrite policy <name>

Example 1. Rewriting HTTP Content:

```
> add rewrite policy policyNew "HTTP.RES.IS_VALID" insertact NOREWRITE
 Done
> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertact
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

Example 2. Rewriting a TCP Payload (TCP Rewrite):

```
> add rewrite policy client_tcp_payload_policy CLIENT.IP.SRC.EQ(172.168.12.232) client_tcp_payload_replace_all
 Done
> show rewrite policy client_tcp_payload_policy
        Name: client_tcp_payload_policy
        Rule: CLIENT.IP.SRC.EQ(172.168.12.232)
        RewriteAction: client_tcp_payload_replace_all
        UndefAction: Use Global
        LogAction: Use Global
        Hits: 0
        Undef Hits: 0

 Done
>
<!--NeedCopy-->
```

To modify an existing rewrite policy by using the command line interface

At the command prompt, type the following commands to modify an existing rewrite policy and verify the configuration:

- <set rewrite policy <name> -rule <expression> -action <action> [<undefaction>]
- <show rewrite policy <name>

Example:

```
> set rewrite policy policyNew -rule "HTTP.RES.IS_VALID" -action
   insertaction
 Done

> show rewrite policy policyNew
        Name: policyNew
        Rule: HTTP.RES.IS_VALID
        RewriteAction: insertaction
        UndefAction: NOREWRITE
        Hits: 0
        Undef Hits: 0

 Done
<!--NeedCopy-->
```

To remove a rewrite policy by using the command line interface

At the command prompt, type the following command to remove a rewrite policy:

rm rewrite policy <name>

Example:

```
> rm rewrite policy policyNew
Done
<!--NeedCopy-->
```

To configure a rewrite policy by using the GUI

1. <Navigate to AppExpert > Rewrite > Policies.
2. <In the details pane, do one of the following:
    - <To create a new policy, click Add.
    - <To modify an existing policy, select the policy, and then click Open.
3. <Click Create or OK. A message appears in the status bar, stating that the Policy has been configured successfully.
4. <Repeat steps 2 through 4 to create or modify as many rewrite actions as you wish.
5. <Click Close. To delete a rewrite policy, select the rewrite policy you want to delete, then click Remove and, when prompted, confirm your choice by clicking OK.

Binding a Rewrite Policy

September 21, 2020

After creating a rewrite policy, you must bind it to put it into effect. You can bind your policy to Global if you want to apply it to all traffic that passes through your NetScaler, or you can bind your policy to a specific virtual server or bind point to direct only that virtual server or bind point's incoming traffic to that policy. If an incoming request matches a rewrite policy, the action associated with that policy is carried out.

Rewrite policies for evaluating HTTP requests and responses can be bound to virtual servers of type HTTP or SSL, or they can be bound to the
REQ_OVERRIDE,
REQ_DEFAULT,
RES_OVERRIDE, and
RES_DEFAULT bind points. Rewrite policies for TCP rewrite can be bound only to virtual servers of type
TCP or
SSL_TCP, or to the
OTHERTCP_REQ_OVERRIDE,
OTHERTCP_REQ_DEFAULT,
OTHERTCP_RES_OVERRIDE, and
OTHERTCP_RES_DEFAULT bind points.

Note: The term
OTHERTCP is used in the context of the NetScaler appliance to refer to all
TCP or
SSL_TCP requests and responses that you want to treat as a raw stream of bytes regardless of the protocols that the TCP packets encapsulate.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order - the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is applied first, then the policy assigned a priority of 100, and finally the policy assigned an order of 1000.

Unlike most other features in the NetScaler operating system, the rewrite feature continues to evaluate and implement policies after a request matches a policy. However, the effect of a particular action policy on a request or response will often be different depending on whether it is performed before or after another action. Priority is important to get the results you intended.

You can leave yourself plenty of room to add other policies in any order, and still set them to evaluate in the order you want, by setting priorities with intervals of 50 or 100 between each policy when you bind it. If you do this, you can add additional policies at any time without having to reassign the priority of an existing policy.

When binding a rewrite policy, you also have the option of assigning a goto expression (gotoPriorityExpression) to the policy. A goto expression can be any positive integer that matches the priority assigned to a different policy that has a higher priority than the policy that contains the goto expression. If you assign a goto expression to a policy, and a request or response matches the policy, the NetScaler will immediately go to the policy whose priority matches the goto expression. It will skip over any policies with priority numbers that are lower than that of the current policy, but higher than the priority number of the goto expression, and not evaluate those policies.

To globally bind a rewrite policy by using the command line interface

At the command prompt, type the following commands to globally bind a rewrite policy and verify the configuration:

- bind rewrite global <policyName> <priority> [<gotoPriorityExpression>] [-type <type>] [-invoke (<labelType> <labelName>)]
- show rewrite global

Example:

```
>bind rewrite global policyNew 10
 Done

> show rewrite global
1)       Global bindpoint: RES_DEFAULT
         Number of bound policies: 1

2)       Global bindpoint: REQ_OVERRIDE
         Number of bound policies: 1

 Done
<!--NeedCopy-->
```

To bind rewrite policy to a specific virtual server by using the command line interface

At the command prompt, type the following commands to bind rewrite policy to a specific virtual server and verify the configuration:

bind lb vserver <name>@ (<serviceName>@ [-weight <positive_integer>])	<serviceGroupName>@	(-policyName <string>@ [-priority <positive_integer>] [-gotoPriorityExpression <expression>] [-type (REQUEST	RESPONSE)] [-invoke (<labelType> <labelName>)])

-
- show lb vserver <name>

Example:

```
> bind lb vserver lbvip -policyName ns_cmp_msapp -priority 50
 Done
>
> show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+226 ms)
        Time since last state change: 28 days, 01:57:26.350
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:50
2)      Policy : cf-pol Priority:1      Inherited
 Done
<!--NeedCopy-->
```

To bind a rewrite policy to a bind point by using the GUI

1. Navigate to AppExpert > Rewrite > Policies.

2. In the details pane, select the rewrite policy you want to globally bind, and then click Policy Manager.
3. In the Rewrite Policy Manager dialog box, in the Bind Points menu, do one of the following:
    a) If you want to configure bindings for HTTP rewrite policies, click HTTP, and then click either Request or Response, depending on whether you want to configure request-based rewrite policies or response-based rewrite policies.
    b) If you want to configure bindings for TCP rewrite policies, click TCP, and then click either Client or Server, depending on whether you want to configure client-side TCP rewrite policies or server-side TCP rewrite policies.
4. Click the bind point to which you want to bind the rewrite policy. The Rewrite Policy Manager dialog box displays all the rewrite policies that are bound to the selected bind point.
5. Click Insert Policy to insert a new row and display a drop-down list with all available, unbound rewrite policies.
6. Click the policy you want to bind to the bind point. The policy is inserted into the list of rewrite policies bound to the bind point.
7. In the Priority column, you can change the priority to any positive integer. For more information about this parameter, see priority in "Parameters for binding a rewrite policy."
8. If you want to skip over policies and go directly to a specific policy in the event that the current policy is matched, change the value in the Goto Expression column to equal the priority of the next policy to be applied.. For more information about this parameter, see gotoPriorityExpression in "Parameters for binding a rewrite policy."
9. To modify a policy, click the policy, and then click Modify Policy.
10. To unbind a policy, click the policy, and then click Unbind Policy.
11. To modify an action, in the Action column, click the action you want to modify, and then click Modify Action.
12. To modify an invoke label, in the Invoke column, click the invoke label you want to modify, and then click Modify Invoke Label.
13. To regenerate the priorities of all the policies that are bound to the bind point you are currently configuring, click Regenerate Priorities. The policies retain their existing priorities relative to the other policies, but the priorities are renumbered in multiples of ten.
14. Click Apply Changes.
15. Click Close. A message appears in the status bar, stating that the Policy has been configured successfully.

To bind a rewrite policy to a specific virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane list of virtual servers, select the virtual server to which you want to bind the rewrite policy, and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab. All policies

configured on your NetScaler appear on the list.

4. Select the check box next to the name of the policy you want to bind to this virtual server.
5. Click OK. A message appears in the status bar, stating that the Policy has been configured successfully.

Configuring Rewrite Policy Labels

September 21, 2020

If you want to build a more complex policy structure than is supported by single policies, you can create policy labels and then bind them as you would policies. A policy label is a user-defined point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority you configured. A policy label can include one or multiple policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and the return of control to the policy that invoked the policy label.

A rewrite policy label consists of a name, a transform name that describes the type of policy included in the policy label, and a list of policies bound to the policy label. Each policy that is bound to the policy label contains all of the elements described in Configuring a Rewrite Policy.

Note: You can use either the command line interface or the GUI to create and configure rewrite policy labels. Users who are not thoroughly familiar with the command line interface and the NetScaler Policy Infrastructure (PI) language will usually find using the GUI much easier.

To configure a rewrite policy label by using the command line interface

To add a new rewrite policy label, at the command prompt, type the following command:

add rewrite policylabel <labelName> <transform>

For example, to add a rewrite policy label named polLabelHTTPResponses to group all policies that work on HTTP responses, you would type the following:

add rewrite policylabel polLabelHTTPResponses http_res

To modify an existing rewrite policy label, at the NetScaler command prompt, type the following command:

set rewrite policy <name> <transform>

Note: The set rewrite policy command takes the same options as the add rewrite policy command.

To remove a rewrite policy label, at the NetScaler command prompt, type the following command:

rm rewrite policy<name>

For example, to remove a rewrite policy label named polLabelHTTPResponses, you would type the following:

rm rewrite policy polLabelHTTPResponses

To configure a rewrite policy label by using the GUI

1. Navigate to AppExpert > Rewrite > Policy Labels.
2. In the details pane, do one of the following:
    - To create a new policy label, click Add.
    - To modify an existing policy label, select the policy, and then click Open.
3. Add or remove policies from the list that is bound to the policy label.
    - To add a policy to the list, click Insert Policy, and choose a policy from the drop-down list. You can create a new policy and add it to the list by choosing New Policy in the list, and following the instructions in "Configuring a Rewrite Policy."
    - To remove a policy from the list, select that policy, and then click Unbind Policy.
4. Modify the priority of each policy by editing the number in the Priority column.

    You can also automatically renumber policies by clicking Regenerate Priorities.
5. Click Create or OK, and then click Close.

    To remove a policy label, select it, and then click Remove. To rename a policy label, select it and then click Rename. Edit the name of the policy, and then click OK to save your changes.

Configuring the Default Rewrite Action

September 21, 2020

An undefined event is triggered when the NetScaler cannot evaluate a policy, usually because it detects a logical or other error in the policy or an error condition on the NetScaler. When the rewrite policy evaluation results in an error, the specified undefined action is carried out. Undefined actions configured at the rewrite policy level are carried out before a globally configured undefined action.

The NetScaler supports following three types of undefined actions:

- undefAction NOREWRITE

  Aborts rewrite processing, but does not alter the packet flow. This means that the NetScaler continues to process requests and responses that do not match any rewrite policy, and eventually forwards them to the requested URL unless another feature intervenes and blocks or redirects the request. This action is appropriate for normal requests to your Web servers, and is the default setting.

- undefAction RESET

  Resets the client connection. This means that the NetScaler tells the client that it must re-establish its session with the Web server. This action is appropriate for repeat requests for Web pages that do not exist, or for connections that might be attempts to hack or probe your protected Web site(s).

- undefAction DROP

  Silently drops the request without responding to the client in any way. This means that the NetScaler simply discards the connection without responding to the client. This action is appropriate for requests that appear to be part of a DDoS attack or another sustained attack on your servers.

Note: Undefined events can be triggered for both request and response flow specific policies.

To configure the default action by using the command line interface

At the command prompt, type the following commands to configure the default action and verify the configuration:

<set rewrite param -undefAction (NOREWRITE	RESET	DROP)

-
- <show rewrite param

Example:

```
> set rewrite param -undefAction NOREWRITE
 Done
> show rewrite param
        Action Name: NOREWRITE
 Done
<!--NeedCopy-->
```

To configure the default action by using the GUI

1. Navigate to AppExpert > Rewrite.

2. In the details pane, under Rewrite Overview, click the Change Rewrite Settings link. The Set Rewrite Params dialog box appears.
3. Under Global Undefined-Result Action, select an option as follows:
    - NoRewrite—NOREWRITE
    - Reset—RESET
    - Drop—DROP
4. Click OK. The global undefined action is set to the value you chose.

Bypassing the Safety Check

September 21, 2020

When you create a rewrite action, the NetScaler verifies that the expression you used to create the action is safe. Expressions created by the NetScaler from run-time data, such as URLs contained in HTTP requests, can cause unexpected errors. The NetScaler reports expressions that cause such errors as unsafe expressions.

In some cases, the expressions may be safe. For example, the NetScaler cannot validate an expression that contains a URL that does not resolve, even if the URL does not resolve because the Web server is temporarily unavailable. You can manually bypass the Safety Check to allow these expressions.

To bypass the safety check by using the command line interface

At the command prompt, type the following commands to bypass the safety check and verify the configuration:

- <set rewrite action <name> -bypassSafetyCheck YES
- <show rewrite action <name>

Example:

```
> set rewrite action insertact -bypassSafetyCheck YES
 Done
> show rewrite action insertact

        Name: insertact
        Operation: insert_http_header   Target:Client-IP
        Value:CLIENT.IP.SRC
        BypassSafetyCheck : YES
        Hits: 0
        Undef Hits: 0
        Action Reference Count: 2
 Done
<!--NeedCopy-->
```

To bypass safety check by using the GUI

1. Navigate to AppExpert > Rewrite > Actions.
2. In the details pane, select the rewrite action to be exempted from the safety check, and then click Open.
3. In the Configure Rewrite Action dialog box, select the Bypass Safety Check check box.
4. Click OK.

Rewrite Action and Policy Examples

September 21, 2020

The examples in this section demonstrate how to configure rewrite to perform various useful tasks. The examples occur in the server room of Example Manufacturing Inc., a mid-sized manufacturing company that uses its Web site to manage a considerable portion of its sales, deliveries, and customer support.

Example Manufacturing has two domains: example.com for its Web site and email to customers, and example.net for its intranet. Customers use the Example Web site to place orders, request quotes, research products, and contact customer service and technical support.

As an important part of Example's revenue stream, the Web site must respond quickly and keep customer data confidential. Example therefore has several Web servers and uses NetScaler appliances to balance the Web site load and manage traffic to and from its Web servers.

The Example system administrators use the rewrite features to perform the following tasks:

Example 1: Delete old X-Forwarded-For and Client-IP Headers

Example Inc. removes old X-Forwarded-For and Client-IP HTTP headers from incoming requests.

Example 2: Adding a Local Client-IP Header

Example Inc. adds a new, local Client-IP header to incoming requests.

Example 3: Tagging Secure and Insecure Connections

Example Inc. tags incoming requests with a header that indicates whether the connection is a secure connection.

Example 4: Mask the HTTP Server Type

Example Inc. modifies the HTTP Server: header so that unauthorized users and malicious code cannot use that header to determine the HTTP server software it uses.

Example 5: Redirect an External URL to an Internal URL

Example Inc. hides information about the actual names of its Web servers and the configuration of its server room from users, to make URLs on its Web site shorter and easier to remember and to improve security on its site.

Example 6: Migrating Apache Rewrite Module Rules

Example Inc. moved its Apache rewrite rules to a NetScaler appliance, translating the Apache PERL-based script syntax to the NetScaler rewrite rule syntax.

Example 7: Marketing Keyword Redirection

The marketing department at Example Inc. sets up simplified URLs for certain predefined keyword searches on the company's Web site.

Example 8: Redirect Queries to the Queried Server.

Example Inc. redirects certain query requests to the appropriate server.

Example 9: Home Page Redirection

Example Inc. recently acquired a smaller competitor, and it now redirects requests to the acquired company's home page to a page on its own Web site.

Each of these tasks requires that the system administrators create rewrite actions and policies and bind them to a valid bind point on the NetScaler.

Example 1: Delete Old X-Forwarded-For and Client-IP Headers

September 21, 2020

Example Inc. wants to remove old X-Forwarded-For and Client-IP HTTP headers from incoming requests, so that the only X-Forwarded-For headers that appear are the ones added by the local server. This configuration can be done through the NetScaler command line or the GUI. The Example Inc. system administrator is an old-school networking engineer and prefers to use a CLI where possible, but wants to be sure he understands the GUI interface so that he can show new system administrators on the team how to use it.

The examples below demonstrate how to perform each configuration with both the CLI and the GUI. The procedures are abbreviated on the assumption that users will already know the basics of creating rewrite actions, creating rewrite policies, and binding policies.

- For more detailed information about creating rewrite actions, see Configuring a Rewrite Action.

- For more detailed information about creating rewrite policies, see Configuring a Rewrite Policy.
- For more detailed information about binding rewrite policies, see Binding a Rewrite Policy.

To delete old X-Forwarded and Client-IP headers from a request by using the command line interface

At the command prompt, type the following commands in the order shown:

```
add rewrite action act_del_xfor delete_http_header x-forwarded-for
add rewrite action act_del_cip delete_http_header client-ip
add rewrite policy pol_check_xfor 'HTTP.REQ.HEADER("x-forwarded-for").
    EXISTS' act_del_xfor
add rewrite policy pol_check_cip 'HTTP.REQ.HEADER("client-ip").EXISTS'
    act_del_cip
bind rewrite global pol_check_xfor 100 200
bind rewrite global pol_check_cip 200 300
<!--NeedCopy-->
```

To delete old X-Forwarded and Client-IP headers from a request by using the GUI

In the Create Rewrite Action dialog box, create two rewrite actions with the following descriptions.

Name	Type	Argument(s)
act_del_xfor	delete_http_header	x-forwarded-for
act_del_cip	delete_http_header	client-ip

In the Create Rewrite Policy dialog box, create two rewrite policies with the following descriptions.

Name	Expression	Action
pol_check_xfor	‘HTTP.REQ.HEADER(“x-forwarded-for”).EXISTS’	act_del_xfor
pol_check_cip	‘HTTP.REQ.HEADER(“client-ip”).EXISTS’	act_del_cip

Bind both policies to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_check_xfor	100	200
pol_check_cip	200	300

All old X-Forwarded-For and Client-IP HTTP headers are now deleted from incoming requests.

Example 2: Adding a Local Client-IP Header

June 9, 2021

Example Inc. wants to add a local Client-IP HTTP header to incoming requests. This example contains two slightly different versions of the same basic task.

To add a local Client-IP header by using the command line interface

At the command prompt, type the following commands in the order shown:

```
add rewrite action act_ins_client insert_http_header NS-Client 'CLIENT.
    IP.SRC'
add rewrite policy pol_ins_client 'HTTP.REQ.HEADER("x-forwarded-for").
    EXISTS || HTTP.REQ.HEADER("client-ip").EXISTS' act_ins_client
bind rewrite global pol_ins_client 300 END
<!--NeedCopy-->
```

To add a local Client-IP header by using the GUI

In the Create Rewrite Action dialog box, create a rewrite action with the following description.

Name	Type	Argument(s)
act_ins_client	insert_http_header	NS-Client 'CLIENT.IP.SRC'

In the Create Rewrite Policy dialog box, create a rewrite policy with the following description.

Name	Expression	Action		
pol_ins_client	‘HTTP.REQ.HEADE forwarded-for”).EXISTS		HTTP.REQ.HEADE ip”).EXISTS’	act_ins_client

Bind policy to global, assigning the priorities and goto expression values shown below.

Name	Priority	Goto Expression
pol_ins_client	100	Next

Example 3: Tagging Secure and Insecure Connections

September 21, 2020

Example Inc. wants to tag incoming requests with a header that indicates whether or not the connection is a secure connection. This helps the server keep track of secure connections after the NetScaler has decrypted the connections.

To implement this configuration, you would begin by creating rewrite actions with the values shown in the following tables. These actions label connections to port 80 as insecure connections, and connections to port 443 as secure connections.

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_YES	INSERT_HTTP_HEADER	SSL	YES

Action Name	Type of Rewrite Action	Header Name	Value
Action-Rewrite-SSL_NO	INSERT_HTTP_HEADER	SSL	NO

You would then create a rewrite policy with the values shown in the following tables. These policies check incoming requests to determine which requests are directed to port 80 and which are directed to port 443. The policies then add the correct SSL header.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-SSL_YES	Action-Rewrite-SSL_YES	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(443)
Policy-Rewrite-SSL_NO	Action-Rewrite-SSL_NO	NOREWRITE	CLIENT.TCP.DSTPORT.EQ(80)

Finally, you would bind the rewrite policies to NetScaler, assigning the first policy a priority of 200, and the second a priority of 300, and setting the goto expression of both policies to END.

Each incoming connection to port 80 now has an SSL:NO HTTP header added to it and each incoming connection to port 443 has an SSL:YES HTTP header added to it.

Example 4: Mask the HTTP Server Type

September 21, 2020

Example Inc. wants to modify the HTTP Server: header so that unauthorized users and malicious code cannot use the header to identify the software that the HTTP server uses.

To modify the HTTP Server: header, you would create a rewrite action and a rewrite policy with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Server_Mask	REPLACE	HTTP.RES.HEADER(“Ser	“Web Server 1.0”

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Server_Mask	Action-Rewrite-Server_Mask	NOREWRITE	HTTP.RES.IS_VALID

Example commands:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("
Server")"\"Web Server 1.0\""
```

```
> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-Rewrite-Server_Mask NOREWRITE
```

You would then globally bind the rewrite policy, assigning a priority of 100 and setting the Goto Priority Expression of the policy to END.

The HTTP Server: header is now modified to read "Web Server 1.0," masking the actual HTTP server software used by the Example Inc. Web site.

Example 5: Redirect an External URL to an Internal URL

September 21, 2020

Example Inc. wants to hide its actual server room configuration from users to improve security on its Web servers.

To do this, you would create a rewrite action with the values as shown in the following tables. For request headers, the action in the table modifies www.example.com to web.hq.example.net. For response headers, the action does the opposite, translating web.hq.example.net to www.example.com.

Action Name	**Type of Rewrite Action**	**Expression to choose target reference**	**String expression for replacement text**
Action-Rewrite-Request_Server_Replace	REPLACE	HTTP.REQ.HOSTNAME.SERVER	"Web.hq.example.net"
Action-Rewrite-Response_Server_Replace	REPLACE	HTTP.RES.HEADER("Server")	"www.example.com"

The first policy checks incoming requests to see if they are valid, and if they are, it performs the Action-Rewrite-Request_Server_Replace action. The second policy checks responses to see if they originate at the server web.hq.example.net. If they do, it performs the Action-Rewrite-Response_Server_Replace action.

Examples of rewrite action and policy for redirecting an external URL.

```
add rewrite action Action-Rewrite-Request_Server_Replace REPLACE HTTP.REQ.HOSTNAME.SERVER '"Web.hq.example.net"
```

```
add rewrite action Action-Rewrite-Response_Server_Replace REPLACE HTTP.RES.HEADER("Server")'"www.example.com"'
```

```
add rewrite policy Policy-Rewrite-Request_Server_Replace HTTP.REQ.HOSTNAME.SERVER.EQ("www.example.com")Action-Rewrite-Request_Server_Replace NOREWRITE
add rewrite policy Policy-Rewrite-Response_Server_Replace HTTP.REQ.HEADER("
Server").EQ("Web.hq.example.net")Action-Rewrite-Response_Server_Replace
```

Finally, you would bind the rewrite policies, assigning each a priority of 500 because they are in different policy banks and therefore will not conflict. You should set the goto expression to NEXT for both bindings.

```
bind rewrite global Policy-Rewrite-Request_Server_Replace 500 END -type
REQ_DEFAULT
```

```
bind rewrite global Policy-Rewrite-Response_Server_Replace 500 END -type
RES_DEFAULT
```

All instances of `www.example.com` in the request headers are now changed to `web.hq.example.net`, and all instances of `web.hq.example.net` in response headers are now changed to `www.example.com`.

Example 6: Migrating Apache Rewrite Module Rules

September 21, 2020

Example Inc., is currently using the Apache rewrite module to process search requests sent to its Web servers and redirect those requests to the appropriate server on the basis of information in the request URL. Example Inc. wants to simplify its setup by migrating these rules onto the NetScaler platform.

Several Apache rewrite rules that Example currently uses are shown below. These rules redirect search requests to a special results page if they do not have a SiteID string or if they have a SiteID string equal to zero (0), or to the standard results page if these conditions do not apply.

The following are the current Apache rewrite rules:

- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} !SiteId= [OR]
- RewriteCond %{QUERY_STRING} SiteId=0
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- RewriteRule ^.*$ /results2.html [P,L]
- RewriteCond %{REQUEST_FILENAME} ^/search$ [NC]
- RewriteCond %{QUERY_STRING} CallName=DisplayResults [NC]
- <RewriteRule ^.*$ /results.html [P,L]

To implement these Apache rewrite rules on the NetScaler, you would create rewrite actions with the values in the following tables.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Display_Results_NulSit	REPLACE	HTTP.REQ.URL	"/results2.html"
Action-Rewrite-Display_Results	REPLACE	HTTP.REQ.URL	"/results2.html"

You would then create rewrite policies with the values as shown in the tables below.

Policy Name	Action Name	Undefined Action	Expression		
Policy-Rewrite-Display_Re	Action-Rewrite-Display_Re	NOREWRITI	HTTP.REQ.U && (!HTTP.REQ	HTTP.REQ.U	HTTP.REQ.URL.QUERY.S
Policy-Rewrite-Display_Results	Action-Rewrite-Display_Results	NOREWRITE	HTTP.REQ.URL.PATH.SET_TEXT_MODE(IGNORECASE).EQ("/search")	HTTP.REQ.URL.QUERY.SET_TEXT_MODE(IGNORE	

Finally, you would bind the rewrite policies, assigning the first a priority of 600 and the second a priority of 700, and then set the goto expression to NEXT for both bindings.

The NetScaler now handles these search requests exactly as the Web server did before the Apache rewrite module rules were migrated.

Example 7: Marketing Keyword Redirection

September 21, 2020

The marketing department at Example Inc. wants to set up simplified URLs for certain predefined keyword searches on the company's Web site. For these keywords, it wants to redefine the URL as shown below.

- External URL:

```
http://www.example.com/\<marketingkeyword\>
```

- Internal URL:

```
http://www.example.com/go/kwsearch.asp?keyword=\<marketingkeyword\>
```

To set up redirection for marketing keywords, you would create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target location	String expression for replacement text
Action-Rewrite-Modify_URL	INSERT_BEFORE	HTTP.REQ.URL.PATH.GE	””go/kwsearch.aspkeyword=”l”

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Modify_URL	Action-Rewrite-Modify_URL	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“

Finally, you would bind the rewrite policy, assigning it a priority of 800. Unlike the previous rewrite policies, this policy should be the last to be applied to a request that matches its criteria. For this reason, NetScaler administrator sets its Goto Priority Expression to END.

Any request using a marketing keyword is redirected to the keyword search CGI page, whereupon a search is performed and all remaining policies are skipped.

Example 8: Redirect Queries to the Queried Server

September 21, 2020

Example Inc. wants to redirect query requests to the appropriate server, as shown here.

- `<Request: GET /query.cgi?server=5HOST: www.example.com`
- `<Redirect URL: <http://web-5.example.com/>`

To implement this redirection, you would first create a rewrite action with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_Hostheader	REPLACE	`HTTP.REQ.HEADER("Host").BEFORE\\_STR(".example.com")`	“server-“ + HTTP.REQ.URL.QUERY.VALUE(“web”)

You would then create a rewrite policy with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace_Hostheader	Action-Rewrite-Replace_Hostheader	NOREWRITE	`HTTP.REQ.HEADER("Host").EQ("www.example.com")`

Example CLI commands:

```
> add rewrite action Action-Rewrite-Server_Mask REPLACE HTTP.RES.HEADER("
Server")"\"Web Server 1.0\""
Done
```

```
> add rewrite policy Policy-Rewrite-Server_Mask HTTP.RES.IS_VALID Action-
Rewrite-Server_Mask NOREWRITE
Done
```

Finally, you would bind the rewrite policy, assigning it a priority of 900. Because this policy should be the last policy applied to a request that matches its criteria, you set the goto expression to END.

Incoming requests to any URL that begins with `<http://www.example.com/query.cgi?server>=` are redirected to the server number in the query.

Example 9: Home Page Redirection

September 21, 2020

New Company, Inc. recently acquired a smaller competitor, Purchased Company, and wants to redirect the home page for Purchased Company to a new page on its own Web site, as shown here.

- Old URL: http://www.purchasedcompany.com/*
- New URL: http://www.newcompany.com/products/page.htm

To redirect requests to the Purchased Company home page, you would create rewrite actions with the values in the following table.

Action Name	Type of Rewrite Action	Expression to choose target reference	String expression for replacement text
Action-Rewrite-Replace_URLr	REPLACE	HTTP.REQ.URL.PATH_A	“/products/page.htm”
Action-Rewrite-Replace_Host	REPLACE	HTTP.REQ.HOSTNAME	“www.newcompany.com”

You would then create rewrite policies with the values in the following table.

Policy Name	Action Name	Undefined Action	Expression
Policy-Rewrite-Replace-None	Action-Rewrite-Replace-None	NOREWRITE	!HTTP.REQ.HOSTNAME.SERVER.EQ(“
Policy-Rewrite-Replace-Host	Action-Rewrite-Replace_Host	NOREWRITE	HTTP.REQ.HOSTNAME.SERVER.EQ(“v
Policy-Rewrite-Replace-URL	Action-Rewrite-Replace_URL	NOREWRITE	HTTP.REQ.IS_VALID

Finally, you would bind the rewrite policies globally, assigning the first a priority of 100, the second a priority of 200, and the third a priority of 300. These policies should be the last policies applied to a request that matches the criteria. For this reason, set the goto expression to END for the first and third policies, and to 300 for the second policy. This ensures that all remaining requests are processed correctly.

Requests to the acquired company’s old Web site are now redirected to the correct page on the New Company home page.

URL Transformation

September 21, 2020

The URL transformation feature provides a method for modifying all URLs in designated requests from an external version seen by outside users to an internal URL seen only by your Web servers and IT staff. You can redirect user requests seamlessly, without exposing your network structure to users. You can also modify complex internal URLs that users may find difficult to remember into simpler, more easily remembered external URLs.

> **Note**
>
> Before you can use the URL transformation feature, you must enable the Rewrite feature. To enable the Rewrite feature, see Enabling the Rewrite Feature.
>
> URL transform feature rewrites URLs in HTML response body and is not applied to JavaScript and other variables.

To begin configuring URL transformation, you create profiles, each describing a specific transformation. Within each profile, you create one or more actions that describe the transformation in detail. Next, you create policies, each of which identifies a type of HTTP request to transform, and you associate each policy with an appropriate profile. Finally, you globally bind each policy to put it into effect.

Configuring URL Transformation Profiles

September 21, 2020

A profile describes a specific URL transformation as a series of actions. The profile functions primarily as a container for the actions, determining the order in which the actions are performed. Most transformations transform an external hostname and optional path into a different, internal hostname and path. Most useful transformations are simple and require only a single action, but you can use multiple actions to perform complex transformations.

You cannot create actions and then add them to a profile. You must create the profile first, and then add actions to it. In the CLI, creating an action and configuring the action are separate steps. Creating a profile and configuring the profile are separate steps in both the CLI and the GUI.

To create a URL transformation profile by using the NetScaler command line

At the NetScaler command prompt, type the following commands, in the order shown, to create a URL transformation profile and verify the configuration. You can then repeat the second and third commands to configure additional actions:

```
add transform profile <profileName> -type URL     OFF)] [-comment <comment>]
[-onlyTransformAbsURLinBody (ON
```

-

- add transform action <name> <profileName> <priority>

set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainFrom <expression>] [-cookieDomainInto <expression>] [-state (ENABLED	DISABLED)] [-comment “<string>”]

-
- show transform profile <name>

Example:

```
> add transform profile shoppingcart -type URL
 Done
> add transform action actshopping shoppingcart 1000
 Done
> set transform action actshopping -priority 1000 -reqUrlFrom 'shopping.example.com' -reqUrlInto 'www.example.net/shopping' -resUrlFrom 'www.example.net/shopping' -resUrlInto 'shopping.example.com' -cookieDomainFrom 'example.com' -cookieDomainInto 'example.net' -state ENABLED -comment 'URL transformation for shopping cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

To modify an existing URL transformation profile or action by using the NetScaler command line

At the NetScaler command prompt, type the following commands to modify an existing URL transformation profile or action and verify the configuration:

Note: Use a set transform profile or set transform action command, respectively. The set transform profile command takes the same arguments as does the add transform profile command, and set transform action is the same command that was used for initial configuration.

set transform action <name> [-priority <priority>] [-reqUrlFrom <expression>] [-reqUrlInto <expression>] [-resUrlFrom <expression>] [-resUrlInto <expression>] [-cookieDomainInto <expression>] [-state (ENABLED	DISABLED)] [-comment “<string>”]

-
- show transform profile <name>

Example:

```
> set transform action actshopping -priority 1000 -reqUrlFrom 'searching.example.net' -reqUrlInto 'www.example.net/searching' -resUrlFrom 'www.example.net/searching' -resUrlInto 'searching.example.com' -cookieDomainInto 'example.net' -state ENABLED -comment 'URL transformation for searching cart.'
 Done
> show transform profile shoppingcart
        Name: shoppingcart
                Type: URL       onlyTransformAbsURLinBody: OFF
        Comment:
        Actions:

1)              Priority 1000   Name: actshopping       ENABLED
 Done
<!--NeedCopy-->
```

To remove a URL transformation profile and actions by using the NetScaler command line

First remove all actions associated with that profile by typing the following command once for each action:

- rm transform action <name> After you have removed all actions associated with a profile, remove the profile as shown below.
- rm transform profile <name>

To create a URL transformation profile by using the GUI

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Profiles.
2. In the details pane, click Add.
3. In the Create URL Transformation Profile dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
    - Name*—name
    - Comment—comment
    - Only transform absolute URLs in response body—onlyTransformAbsURLinBody
4. Click Create, and then click Close. A message appears in the status bar, stating that the Profile has been configured successfully.

To configure a URL transformation profile and actions by using the GUI

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Profiles.
2. In the details pane, select the profile you want to configure, and then click Open.
3. In the Configure URL Transformation Profile dialog box, do one of the following.
    - To create a new action, click Add.
    - To modify an existing action, select the action, and then click Open.
4. Fill in the Create URL Transformation Action or Modify URL Transformation Action dialog box by typing or selecting values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation profiles" as follows (asterisk indicates a required parameter):
    - Action Name*—name
    - Comments—comment
    - Priority*—priority
    - Request URL from—reqUrlFrom
    - Request URL into—reqUrlInto
    - Response URL from—resUrlFrom
    - Response URL into—resUrlInto
    - Cookie Domain from—cookieDomainFrom
    - Cookie Domain into—cookieDomainInto
    - Enabled—state
5. Save your changes.
    - If you are creating a new action, click Create, and then Close.
    - If you are modifying an existing action, click OK.
      A message appears in the status bar, stating that the Profile has been configured successfully.
6. Repeat step 3 through step 5 to create or modify any additional actions.

7. To delete an action, select the action, and then click Remove. When prompted, click OK to confirm the deletion.
8. Click OK to save your changes and close the Modify URL Transformation Profile dialog box.
9. To delete a profile, in the details pane select the profile, and then click Remove. When prompted, click OK to confirm the deletion.

Configuring URL Transformation Policies

September 21, 2020

After you create a URL transformation profile, you next create a URL transformation policy to select the requests and responses that the NetScaler should transform by using the profile. URL transformation considers each request and the response to it as a single unit, so URL transformation policies are evaluated only when a request is received. If a policy matches, the NetScaler transforms both the request and the response.

Note: The URL transformation and rewrite features cannot both operate on the same HTTP header during request processing. Because of this, if you want to apply a URL transformation to a request, you must make sure that none of the HTTP headers it will modify are manipulated by any rewrite action.

To configure a URL transformation policy by using the NetScaler command line

You must create a new policy. On the command line, an existing policy can only be removed. At the NetScaler command prompt, type the following commands to configure a URL transformation policy and verify the configuration:

- <add transform policy <name> <rule> <profileName>
- <show transform policy <name>

Example:

```
> add transform policy polsearch  HTTP.REQ.URL.SUFFIX.EQ("Searching")
   prosearching
 Done
> show transform policy polsearch
1)      Name: polsearch
        Rule: HTTP.REQ.URL.SUFFIX.EQ("Searching")
        Profile: prosearching
        Priority: 0
        Hits: 0
 Done
```

```
<!--NeedCopy-->
```

To remove a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following command to remove a URL transformation policy:

rm transform policy <name>

Example:

```
> rm transform policy polsearch
Done
<!--NeedCopy-->
```

To configure a URL transformation policy by using the GUI

1. In the navigation pane, expand Rewrite, expand URL Transformation, and then click Policies.
2. In the details pane, do one of the following:
    - To create a new policy, click Add.
    - To modify an existing policy, select the policy, and then click Open.
3. In the Create URL Transformation Policy or Configure URL Transformation Policy dialog box, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring URL transformation policies" as follows (asterisk indicates a required parameter):
    - Name*—name (Cannot be changed for a previously configured policy.)
    - Profile*—profileName
    - Expression—rule

    If you want help with creating an expression for a new policy, you can either hold down the Control key and press the space bar while your cursor is in the Expression text box. To create the expression, you can type it directly as described below, or you can use the Add Expression dialog box.

    a) Click Prefix, and choose the prefix for your expression.

    Your choices are:

    - HTTP—The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
    - SYS—The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

        - CLIENT—The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
        - SERVER—The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.
        - URL—The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.
        - TEXT—Any text string in the request. Choose this if you want to examine a text string in the request.
        - TARGET—The target of the request. Choose this if you want to examine some aspect of the request target.

          After you choose a prefix, the NetScaler displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom. The choices depend on which prefix you chose.

    b) Select your next term.

       If you chose HTTP as your prefix, your choices are REQ, which specifies HTTP requests, and RES, which specifies HTTP responses. If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

       When you are certain which choice you want, double-click it to insert it into the Expression window.

    c) Type a period, and then continue selecting terms from the list boxes that appear to the right of the previous list box. You type the appropriate text strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

4. Click Create or OK, depending on whether you are creating a new policy or modifying an existing policy.
5. Click Close. A message appears in the status bar, stating that the Policy has been configured successfully.

To add an expression by using the Add Expression dialog box

1. In the Create Responder Action or Configure Responder Action dialog box, click Add.
2. In the Add Expression dialog box, in the first list box choose the first term for your expression.
    - HTTP

      The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.

- SYS

  The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.

- CLIENT

  The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

- SERVER

  The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

- URL

  The URL of the request. Choose this if you want to examine some aspect of the URL to which the request was sent.

- TEXT

  Any text string in the request. Choose this if you want to examine a text string in the request.

- TARGET

  The target of the request. Choose this if you want to examine some aspect of the request target.
  When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.

3. <In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
4. <Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Globally Binding URL Transformation Policies

September 21, 2020

After you have configured your URL transformation policies, you bind them to Global or a bind point to put them into effect. After binding, any a request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the URL transformation feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you tell the NetScaler to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you tell the NetScaler to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you globally bind your policies.

Note: URL transformation policies cannot be bound to TCP-based virtual servers.

To bind a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a URL transformation policy and verify the configuration:

- `bind transform global \<policyName\> \<priority\>`
- `show transform global`

Example:

```
> bind transform global polisearching 100
 Done
> show transform global
1)      Policy Name: polisearching
        Priority: 100

 Done
<!--NeedCopy-->
```

To bind a URL transformation policy by using the GUI

1. In the navigation pane, expand Rewrite, then expand URL Transformation, and then click Policies.
2. In the details pane, click Policy Manager.
3. In the Transform Policy Manager dialog box, choose the bind point to which you want to bind the policy. The choices are:
    - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.

    - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
    - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
    - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
    - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
4. Select Insert Policy to insert a new row and display a drop-down list with all available, unbound URL transformation policies.
5. Select the policy you want to bind, or select New Policy to create a new policy. The policy that you selected or created is inserted into the list of globally bound URL transformation policies.
6. Make any additional adjustments to the binding.
    - <To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
    - To modify the policy expression, double click that field to open the Configure Transform Policy dialog box, where you can edit the policy expression.
    - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
    - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression.
7. Repeat steps 3 through 6 to add any additional URL transformation policies you want to globally bind.
8. Click OK to save your changes. A message appears in the status bar, stating that the Policy has been configured successfully.

Globally Binding URL Transformation Policies

September 21, 2020

After you have configured your URL transformation policies, you bind them to Global or a bind point to put them into effect. After binding, any a request or response that matches a URL transformation policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the URL transformation feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you tell the NetScaler to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you tell the NetScaler to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you globally bind your policies.

Note: URL transformation policies cannot be bound to TCP-based virtual servers.

To bind a URL transformation policy by using the NetScaler command line

At the NetScaler command prompt, type the following commands to globally bind a URL transformation policy and verify the configuration:

- `bind transform global \<policyName\> \<priority\>`
- `show transform global`

Example:

```
> bind transform global polisearching 100
 Done
> show transform global
1)      Policy Name: polisearching
        Priority: 100

 Done
<!--NeedCopy-->
```

To bind a URL transformation policy by using the GUI

1. In the navigation pane, expand Rewrite, then expand URL Transformation, and then click Policies.
2. In the details pane, click Policy Manager.
3. In the Transform Policy Manager dialog box, choose the bind point to which you want to bind the policy. The choices are:
    - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.

  - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
  - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
  - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
  - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
4. Select Insert Policy to insert a new row and display a drop-down list with all available, unbound URL transformation policies.
5. Select the policy you want to bind, or select New Policy to create a new policy. The policy that you selected or created is inserted into the list of globally bound URL transformation policies.
6. Make any additional adjustments to the binding.
   - <To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure Transform Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression.
7. Repeat steps 3 through 6 to add any additional URL transformation policies you want to globally bind.
8. Click OK to save your changes. A message appears in the status bar, stating that the Policy has been configured successfully.

Diameter Support for Rewrite

September 21, 2020

The Rewrite feature now supports the Diameter protocol. You can configure Rewrite to modify Diameter requests and response as you would HTTP or TCP requests and responses, allowing you to use Rewrite to manage the flow of Diameter requests and make necessary modifications. For example, if the “Origin-Host” value in a Diameter request is inappropriate, you can use Rewrite to replace it with a value that is acceptable to the Diameter server.

To configure Rewrite to modify a Diameter request

To configure the Rewrite feature to replace the Origin-Host in a diameter request with a different value, at the command prompt, type the following commands:

- <add rewrite action <actname> replace “DIAMETER.REQ.AVP(264,\“netscaler.example.net\”)”
  For <actname>, substitute a name for your new action. The name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For NetScaler.example.net, substitute the Host-Origin that you want to use instead of the original Host-Name.
- add rewrite policy <polname> “diameter.req.avp(264).value.eq(\“host.example.com\”)” <actname>
  For <polname>, substitute a name for your new policy. As with <actname>, the name can consist of from one to 127 characters in length, and can contain letters, numbers, and the hyphen (-) and underscore (_) symbols. For host.example.com, substitute the name of the Host-Origin that you want to change. For <actname>, substitute the name of the action that you just created.
- bind lb vserver <vservername> -policyName <polname> -priority <priority> -type REQUEST
  For <vservername>, substitute the name of the load balancing virtual server to which you want to bind the policy. For <polname>, substitute the name of the policy you just created. For <priority>, substitute a priority for the policy.

Example:

To create a Rewrite action and policy to modify all Diameter Host-Origins of “host.example.com” to “NetScaler.example.net”, you could add the following action and policy, and bind the policy as shown.

```
> add rewrite action rw_act_replace_avp replace "diameter.req.avp(264)"
   "diameter.new.avp(264,\"netscaler.example.net\")"
> add rewrite policy rw_diam_pol "diameter.req.avp(264).value.eq(\"
  client.realm2.net\")" rw_act_replace_avp
> bind lb vserver vs1 -policyName rw_diam_pol -priority 10 -type
  REQUEST

Done
<!--NeedCopy-->
```

DNS Support for the Rewrite Feature

September 21, 2020

You can configure the rewrite feature to modify DNS requests and responses, as you would for HTTP or TCP requests and responses. You can use rewrite to manage the flow of DNS requests, and make necessary modifications in the header, or in the answer section. For example, if the DNS response does not have the AA bit set in the header flag, you can use rewrite to set the AA bit in the DNS response and send it to the client.

DNS Expressions

In a rewrite configuration, you can use the following NetScaler expressions to refer to various portions of a DNS request or response:

See Expressions and Descriptions

DNS Bind Points

The following global bind points are available for policies that contain DNS expressions.

Bind Points	Description
DNS_REQ_OVERRIDE	Override request policy queue.
DNS_REQ_DEFAULT	Standard request policy queue.
DNS_RES_OVERRIDE	Override response policy queue.
DNS_RES_DEFAULT	Standard response policy queue.

In addition to the default bind points, you can create policy labels of type DNS_REQ or DNS_RES and bind DNS policies to them.

Rewrite Action Types for DNS

- **replace_dns_answer_section**—This action replaces the DNS answers section with the defined expression in the DNS policy.
- **replace_dns_header_field**—Checks the opcode type in the DNS request. Returns True or False, indicating whether the opcode type in the DNS request matches the specified opcode type. This action replaces the DNS header section with the defined expression in the DNS policy.

Configuring Rewrite Policies for DNS

The following procedure uses the NetScaler command line to configure a rewrite action and policy and bind the policy to a rewrite-specific global bind point.

Configure Rewrite action and policy, and bind the policy for DNS

At the command prompt, type the following commands:

1. add rewrite action <actName> <actType>

   For <actname>, substitute a name for your new action. The name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actType>, specify the rewrite action types provided for DNS expressions.

2. add rewrite policy <polName> <rule> <actName>

   For <polname>, substitute a name for your new policy. For <actname>, the name can be 1 to 127 characters in length, and can contain letters, numbers, hyphen (-), and underscore (_) symbols. For <actname>, substitute the name of the action that you just created.

3. bind rewrite global <polName> <priority> < gotoPriorityExpression> -type <bindPoint>

   For <polName>, substitute the name of the policy that you just created. For <priority>, specify the priority of the policy. For <bindPoint>, substitute one of the rewrite -specific global bind points.

Example:

Set the AA bit of DNS request to load balance virtual server.

The following commands configure the NetScaler appliance to act as an authoritative DNS server for all the queries that it serves.

```
add rewrite action set_aa replace_dns_header_field dns.req.header.flags.set(aa)
add rewrite policy pol !dns.req.header.flags.is_set(aa)  set_aa
bind rewrite global  pol  100  -type dns_res_override
<!--NeedCopy-->
```

Modify the response answer and header section.

If the server responds with an NX domain, you can set the rewrite action to replace the response with specified IP address. A NOPOLICY-REWRITE enables you to invoke an enternal bank without processing an expression (a rule). This entry is a dummy policy that does not contain a rule but directs the entry to a policy label or virtual server specific policy banks.

```
add rewrite action set_aa_res replace_dns_header_field "dns.res.header.flags.set(aa)"
add rewrite action modify_nxdomain_res replace_dns_answer_section "dns.new_rrset_a(\"10.102.218.160\",300)"
add rewrite policy set_res_aa true set_aa_res
add add rewrite policy modify_answer "dns.RES.HEADER.RCODE.EQ(nxdomain) && dns.RES.QUESTION.TYPE.EQ(A)"
modify_nxdomain_res
add rewrite policylabel MODIFY_NODATA dns_res
bind rewrite policylabel MODIFY_NODATA modify_answer 10 END
bind rewrite policylabel MODIFY_NODATA set_res_aa 11 END
bind lb vserver v1 -policyName NOPOLICY-REWRITE -priority 11 -gotoPriorityExpression END -type
RESPONSE -invoke policylabel MODIFY_NODATA
<!--NeedCopy-->
```

Limitations:

- Rewrite policies are evaluated only if the NetScaler appliance is configured as a DNS proxy server and there is a cache miss.
- If the Recursion Available (RA) flag in the header is set to YES, the RA flag will not be modified in the rewrites.
- If the RA flag in the header is set to YES, the CD flag in the header is modified regardless of any rewrite action.

String maps

September 21, 2020

You can use string maps to perform pattern matching in all NetScaler features that use the default policy syntax. A string map is a NetScaler entity that consists of key-value pairs. The keys and values are strings in either ASCII or UTF-8 format. String comparison uses two new functions, MAP_STRING() and IS_STRINGMAP_KEY().

A policy configuration that uses string maps performs better than one that does string matching through policy expressions, and you need fewer policies to perform string matching with a large number of key-value pairs. String maps are also intuitive, simple to configure, and result in a smaller configuration.

How String Maps Work

String maps are similar in structure to pattern sets (a pattern set defines a mapping of index values to strings; a string map defines a mapping of strings to strings) and the configuration commands for string maps (commands such as add, bind, unbind, remove, and show) are syntactically similar to configuration commands for pattern sets. Also, as with index values in a pattern set, each key in a string map must be unique across the map. The following table illustrates a string map called url_string_map, which contains URLs as keys and values.

Key	Value
/url_1.html	`http://www.redirect_url_1.com/url_1.html`
/url_2.html	`http://www.redirect_url_2.com/url_2.html`
/url_3.html	`http://www.redirect_url_1.com/url_1.html`

Table 1. String Map "url_string_map"

The following table describes the two functions that have been introduced to enable string matching with keys in a string map. String matching is always performed with the keys. Additionally, the following functions perform a comparison between the keys in the string map and the complete string that is returned by the expression prefix. The examples in the descriptions refer to the preceding example.

For completed information about the two functions introduced for enabling string matching with keys in a string map, see String Map Function table pdf.

Configuring a String Map

You first create a string map and then bind key-value pairs to it. You can create a string map from the command line interface (CLI) or the GUI.

To configure a string map by using the command line interface

At the command prompt, do the following:

1. Create a string map.

   ```
   add policy stringmap <name> -comment <string>
   ```

2. Bind a key-value pair to the string map.

   ```
   bind policy stringmap <name> <key> <value>
   ```

   Example:

```
> bind policy stringmap url_string_map1 "/url_1.html" "http://www.redirect_url_1.com/url_1.html"
<!--NeedCopy-->
```

To configure a string map by using the GUI

Create a string map and bind the key-value pair to the created entity.

Navigate to **AppExpert** > **String Maps**, click **Add** and specify the relevant details.

Example: responder policy with a redirect action

The following use case involves a responder policy with a redirect action. In the example below, the first four commands create the string map url_string_map and bind the three key-value pairs used in the earlier example. After creating the map and binding the key-value pairs, you create a responder action (act_url_redirects) that redirects the client to the corresponding URL in the string map or to www.default.com. You also configure a responder policy (pol_url_redirects) that checks whether requested URLs match any of the keys in url_string_map and then performs the configured action. Finally, you bind the responder policy to the content switching virtual server that receives the client requests that are to be evaluated.

```
add stringmap url_string_map
```

```
bind stringmap url_string_map /url_1.html http://www.redirect_url_1.com/url_1.html
```

```
bind stringmap url_string_map /url_2.html http://www.redirect_url_2.com/url_2.html
```

```
bind stringmap url_string_map /url_3.html http://www.redirect_url_1.com/url_1.html
```

```
add responder action act_url_redirects redirect 'HTTP.REQ.URL.MAP_STRING("url_string_map")ALT "www.default.com"'-bypassSafetyCheck yes
```

```
add responder policy pol_url_redirects TRUE act_url_redirects
```

```
bind cs vserver csw_redirect -policyname pol_url_redirects -priority 1 -type request
```

URL Sets

September 21, 2020

This feature enables you to blacklist one million URLs. The section includes the following topics:

- Getting Started
- Using Advanced Policy Expressions for URL Evaluation
- Configuring a URL Set
- URL Pattern Semantics
- Blacklisted URL Categories

Getting Started

September 21, 2020

To prevent access to restricted websites, a NetScaler appliance uses a specialized URL matching algorithm. The algorithm uses a URL set that can contain a list of URLs up to one million (1,000,000) blacklisted entries. Each entry can include metadata that defines URL categories and category groups as indexed patterns. The appliance can also periodically download URLs of highly sensitive URL sets managed by internet enforcement agencies (with government websites) or independent internet organizations. Once the URL set is downloaded from a website and imported into the appliance, the appliance encrypts the URL sets in the appliance (as required by these agencies) and kept confidential so that the entries are not tampered.

The NetScaler appliance uses advanced policies to determine whether an incoming URL should be blocked, allowed, or redirected. These policies use advanced expressions to evaluate incoming URLs against blacklisted entries. An entry can include metadata. For entries that have no metadata, you might want to use an expression that evaluates the URL on the basis of an exact string match. For other URLs, you might want to use an expression that evaluates the URL's metadata, in addition to an expression that checks for an exact string match.

Use Case for Safe Internet Access Policies for ISPs/Telcos

A URL set enables an Internet Service Provider (ISP) or a Telco customer to enforce government mandated safe internet access policies such as:

1. Block access to illegal internet sites (child abuse, drugs, and so on)
2. Safe browsing for children

A NetScaler appliance enables you to periodically download URL sets managed by internet enforcement agencies or independent internet organizations. The appliance periodically downloads the list and updates it securely. The list is stored as confidential URL sets so that it is not tampered or human readable. The periodically downloaded URL set functions as a blacklisted set for URL evaluation purposes.

If you have a private URL set and the contents of the list are kept confidential and the network administrator does not know about the blacklisted URLs present in the list. To make sure if the policy is configured correctly and the correct list is referenced to evaluate an incoming URL, you configure an internal URL called Canary URL and add it to the URL set. Using the Canary URL, the administrator can request through the appliance use the private URL set to ensure it is looked up for every URL request.

Advanced Policy Expressions for URL Evaluation

September 21, 2020

The following table describes the expressions you can use to evaluate incoming URLs with entries in an URL set.

Note: HTTP.REQ.URL is generalized to be used as <URL expression>

Expression	Operation
<`URL expression`>.URLSET_MATCHES_ANY	Evaluates to TRUE if the URL exactly matches any entry in the URL set.
<`URL expression`> .GET_URLSET_METADATA(<`URLSET`>)	The GET_URLSET_METADATA() expression returns the associated metadata if the URL exactly matches any pattern within the URL set. An empty string is returned if there was no match.
<`URL expression`>.GET_URLSET_METADATA(<`URLSET`>).EQ(<`METADATA`>)	Evaluates to TRUE if the matched metadata is equal to <`METADATA`>.
<`URL expression`> .GET_URLSET_METADATA(<`URLSET`>).	
TYPECAST_LIST_T(‘;’).GET(0).EQ(<`CATEGORY`>)	Evaluates to TRUE if the matched metadata is at the beginning of the category. This pattern can be used to encode separate fields within metadata, but match only the 1st field.
HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL)	Joins the host and URL parameters, which can then be used as a <`URL expression`> for matching.

Configuring URL Set

September 21, 2020

You can perform the following tasks to configure a URL set and restrict URLs on a NetScaler platform:

1.Import a URL set (download and encrypt it). Importing a URL set in a NetScaler appliance allows you to download the URL file, adding the file to the appliance, and then encrypting the file. Until you add the URL set to the system, it will not be visible to the user.

You can download a set in the following ways:

- Download a URL set once from a specific URL using HTTP and HTTPS supported for the file download.
- Download a URL set using FTP.
- Downloading a URL set periodically, using a scheduler that periodically downloads or imports URL sets. The time interval is set in seconds for example, `*http://10.29.102.200/urls.txt -interval 3600*`.

Sample URL Set without a meta data

http://10.102.145.135/top10k

Sample URL Set with a meta data

http://10.102.145.135/blacklists/audio-video/categorized_av

The imported URL set is further categorized into different categories and category groups in the database. This is valid only if categories exist in the metadata of the URL set file.

Note: There can be a chance that you might have URL patterns without metadata.

Once you have downloaded the file, it is pushed into the appliance and at this point of interval, you can update, delete or display file properties. After the file is pushed into the appliance, you can modify the entries by adding further rows as it remains static.

The imported set is then stored in an encrypted file format on the NetScaler directory. The imported list contains millions of URL entries. Otherwise, the appliance returns an error message saying that the value exceeds the limit. If the imported URL set has blacklisted entries with metadata, the metadata it is detected by the appliance when it is imported.

Once you import a URL set and add it into the appliance, the URL set is available for advanced policies to identify the correct URL set during incoming URL evaluation. HTTP.REQ.HOSTNAME.APPEND(HTTP.REQ.URL).UR
`URL set name`>)

2.Updating a URL set on the NetScaler appliance. Once you have pushed the file into the appliance, at this interval you can manually update a URL file by using command line interface.

3.Exporting a URL set. If you prefer a backup of the URL set, you can export the list of URL patterns and save a copy of it to a destination URL. Before you export, check whether the URL set is marked as private. If is marked private, the URL set cannot be exported.

4.Removing a URL set. If you want to delete a URL set of blacklisted entries, you can use the remove command to delete the URL set from the NetScaler appliance.

5.Displaying a URL set. You can display the properties of a URL set by using the show command.

Example:

```
show urlset


Name:  top100           PatternCount: 100        Delimiter:       RowSeparator:           Interval:  0

 Done
<!--NeedCopy-->
```

To import a URL set with meta by using the command line interface

At command prompt, type:

```
import urlset <name> [-overwrite] [ - delimiter <character>] [-rowSeparator <character>] [-url] <url> [-interval <seconds>] [-privateSet] [-canaryUrl <URL>]
<!--NeedCopy-->
```

Where,

delimiter is a CSV file record with default value set as 44.

rowSeparator is a CSV file row separator with default value set as 10.

Interval is the time interval in secs, rounded to the nearest 15 minutes at which the update of urlset occurs.

CanaryURL is a URL used for testing when contents of the urlset is kept confidential.

To show URL set by using the command line interface

At the command prompt, type:

```
show urlset <name>
<!--NeedCopy-->
```

To export an URL set by using the command line interface

At the command prompt, type:

```
export urlset <name> <url>
<!--NeedCopy-->
```

To add an URL set by using the command line interface

At the command prompt, type:

```
add urlset <urlset_name>
<!--NeedCopy-->
```

To update an URL set by using the command line interface

At the command prompt, type:

```
update urlset <name>
<!--NeedCopy-->
```

To remove a URL set command by using the command line interface

At the command prompt, type:

```
remove urlset <name>
<!--NeedCopy-->
```

Example

```
import policy urlset top1k -url http://10.78.79.80/alytra/top1k.csv -delimiter "," -rowSeparator "n" -interval 10 -privateSet -overwrite - canaryUrl http://www.in.gr

add policy urlset top1k

update policy urlset top1k

sh policy urlset

sh policy urlset top1k

```

```
export policy urlset urlset1 -url http://www.example.com/PUT_file_1
import policy urlset top10k -url http://10.102.145.135/top10k -private

add policy urlset top10k

update policy urlset top10k

show policy urlset top10k
<!--NeedCopy-->
```

To import a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, click **Import** to download the URL set.

To add a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, click **Add** to create a URL set file for the downloaded URL set.

To edit a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets**, select a URL set and click **Edit** to modify.

To Update a URL set by using the GUI

Navigate to **AppExpert** > **URL Sets,** select a URL set and click **Update URL Set** to update URL set with the latest modifications made to the file.

To Export a URL set by using the GUI

Navigate to **AppExpert > URL Sets**, select a URL set and click Export URL Set to export the URL patterns in a set to a destination URL and save it in that location.

URL Pattern Semantics

May 5, 2021

The following table shows the URL patterns used for specifying the list of pages you to want to filter. For example, the pattern,`http://www.example.com/bar` matches a single page `http://www.example.com/bar`. To cover all the pages where the URL starts with `www.example.com/bar`, you must explicitly add a '*' at the end.

For more information, see URL pattern metadata mapping table.

URL Categories

September 21, 2020

Following is a list of blacklisted categories.

S.no	Blacklisted Categories
1	Illegal Activities
2	Illegal Drugs
3	Medication
4	Marijuana
5	Terrorism/Extremists
6	Weapons
7	Hate/Slander
8	Violence/Suicide
9	Advocacy in general
10	Adult/Porn
11	Nudity
12	Sexual Services
13	Adult Search/Links
14	Hacking/Cracking
15	Malware
16	Remote Proxies
17	Search Engine Caches
18	Translators
19	Dating
20	Weddings/Matrimony
21	Market Rates
22	Online Trading
23	Insurance

S.no	Blacklisted Categories
24	Financial Products
25	Gambling in general
26	Lottery
27	Online games
28	Games
29	Auctions
30	Shopping/Retail
31	Real Estate
32	IT Online Shopping
33	Web based Chat
34	Instant Messages
35	Web based Mail
36	E-Mail Subscriptions
37	Bulletin Boards
38	IT Bulletin Boards
39	Personal Web Pages/Blogs
40	Downloads
41	Program Downloads
42	Storage Services
43	Streaming Media
44	Employment
45	Career Advancement
46	Side Business
47	Grotesque
48	Special Events
49	Popular Topics
50	Adult Magazine/News
51	Smoking
52	Drinking

S.no	Blacklisted Categories
53	Alcoholic Products
54	Fetish
55	Sexual Expression(text)
56	Costume Play/Enjoyment
57	Occult
58	Home & Family
59	Professional Sports
60	Sports in general
61	Life Events
62	Travel & Tourism
63	Public Agency Tourism
64	Public Transit
65	Accommodations
66	Music
67	Horoscope/Astrology/Fortune Telling
68	Entertainer/Celebrity
69	Dining/Gourmet
70	Entertainment/Venues/Activities
71	Traditional Religions
72	Religions
73	Politics
74	Advertisements/Banners
75	Sweepstakes/Prizes
76	SPAM
77	News
78	Automotive
79	Business & Commercial
80	Computing & Internet
81	Education

S.no	Blacklisted Categories
82	Government
83	Health
84	Internet Telephony
85	Military
86	Peer to Peer/Torrents
87	Recreation & Hobbies
88	Reference
89	Search Engines & Portals
90	Sex Education
91	SMS & Mobile Telephony Services
92	Mobile Apps & Publishers
93	Spyware
94	Content Delivery Networks & Infrastructure
95	Kids Sites
96	Swimsuits & Lingerie
97	Arts & Cultural Events
98	Hosting Sites
99	Philanthropy & Non-Profit Organizations
100	Photo Search & Photo Sharing Sites
101	Ringtones
102	Fashion & Beauty
103	Mobile App Stores
104	Parked Domains
105	Emoticons
106	Mobile Operators
107	Botnets
108	Infected Sites
109	Phishing Sites
110	Keyloggers

S.no	Blacklisted Categories
111	Mobile Malware
112	No Content
113	Agriculture
114	Architecture
115	Associations/Trade Groups/Unions
116	Books/eBooks
117	BOT Phone Home
118	DDNS
119	Unsupported URL
120	Law
121	Local Communities
122	Miscellaneous
123	Online Magazines
124	Pets/Veterinarian
125	Piracy & Copyright Theft
126	Private IP Addresses
127	Recycling/Environment
128	Science
129	Society & Culture
130	Transport Services & Freight
131	Photography & Film
132	Museums & History
133	eLearning
134	Social Networks in General
135	Facebook
136	Facebook: Posting
137	Facebook: Commenting
138	Facebook: Friends
139	Facebook: Photo Upload

S.no	Blacklisted Categories
140	Facebook: Events
141	Facebook: Apps
142	Facebook: Chat
143	Facebook: Questions
144	Facebook: Video Upload
145	Facebook: Groups
146	Facebook: Games
147	LinkedIn
148	LinkedIn: Updates
149	LinkedIn: Mail
150	LinkedIn: Connections
151	LinkedIn: Jobs
152	Twitter
153	Twitter: Posting
154	Twitter: Mail
155	Twitter: Follow
156	YouTube
157	YouTube: Commenting
158	YouTube: Video Upload
159	YouTube: Sharing
160	Instagram
161	Instagram: Upload
162	Instagram: Commenting
163	Instagram: Private Message
164	Tumblr
165	Tumblr: Posting
166	Tumblr: Commenting
167	Tumblr: Photo or Video Upload
168	Google+

S.no	Blacklisted Categories
169	Google+: Posting
170	Google+: Commenting
171	Google+: Photo Upload
172	Google+: Video Upload
173	Google+: Video Chat
174	Pinterest
175	Pinterest: Pin
176	Vine: Upload
177	Vine: Commenting
178	Vine: Message
179	Ask.fm
180	Ask.fm: Ask
181	Ask.fm: Answer
182	YikYak
183	YikYak: Posting
184	YikYak: Commenting
185	Wordpress
186	Wordpress: Posting
187	Wordpress: Upload

AppFlow

September 21, 2020

The NetScaler appliance is a central point of control for all application traffic in the data center. It collects flow and user-session level information valuable for application performance monitoring, analytics, and business intelligence applications. It also collects webpage performance data and database information. AppFlow transmits the information by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. IPFIX (the standardized version of Cisco's NetFlow) is widely used to monitor network flow information. AppFlow defines new Information Elements to represent application-level information, webpage performance data, and database information.

Using UDP as the transport protocol, AppFlow transmits the collected data, called *flow records*, to one or more IPv4 collectors. The collectors aggregate the flow records and generate real-time or historical reports.

AppFlow provides visibility at the transaction level for HTTP, SSL, TCP, and SSL_TCP flows. You can sample and filter the flow types that you want to monitor.

AppFlow use actions and policies to send records for a selected flow to specific set of collectors. An AppFlow action specifies which set of collectors receive the AppFlow records. Policies, which are based on Advanced expressions can be configured to select flows for which flow records are sent to the collectors specified by the associated AppFlow action.

To limit the types of flows, you can enable AppFlow for a virtual server. AppFlow can also provide statistics for the virtual server.

You can also enable AppFlow for a specific service, representing an application server, and monitor the traffic to that application server.

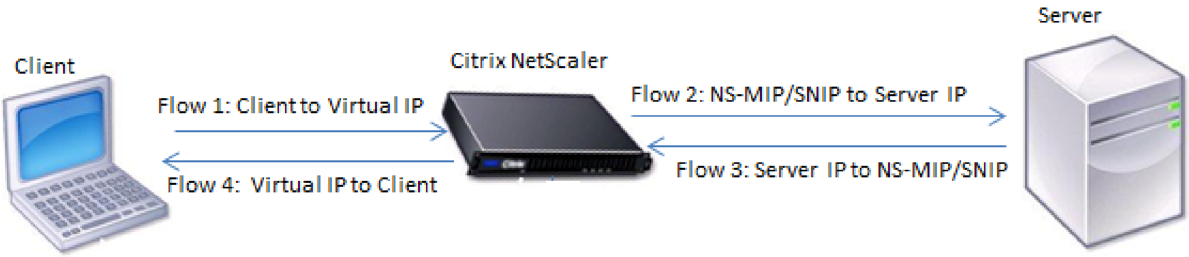
Note: This feature is supported only on NetScaler nCore builds.

How AppFlow Works

In the most common deployment scenario, inbound traffic flows to a Virtual IP address (VIP) on the NetScaler appliance and is load balanced to a server. Outbound traffic flows from the server to a mapped or subnet IP address on the NetScaler and from the VIP to the client. A flow is a unidirectional collection of IP packets identified by the following five tuples: sourceIP, sourcePort, destIP, destPort, and protocol.

The following figure describes how the AppFlow feature works.

Figure 1. NetScaler Flow Sequence


Client
Flow 1: Client to Virtual IP
Flow 4: Virtual IP to Client
Citrix NetScaler
Flow 2: NS-MIP/SNIP to Server IP
Flow 3: Server IP to NS-MIP/SNIP
Server

As shown in the figure, the network flow identifiers for each leg of a transaction depend on the direction of the traffic.

The different flows that form a flow record are:

Flow1: <Client-IP, Client-Port, VIP-IP, VIP-port, Protocol>

Flow2: <NS-MIP/SNIP, NS-port, Server-IP, Server-Port, Protocol>

Flow3: <Server-IP, Server-Port, NS-MIP/SNIP, NS-Port, Protocol>

Flow4: <VIP-IP, VIP-port, Client-IP, Client-Port, Protocol>

To help the collector link all four flows in a transaction, AppFlow adds a custom transactionID element to each flow. For application-level content switching, such as HTTP, it is possible for a single client TCP connection to be load balanced to different back end TCP connections for each request. AppFlow provides a set of records for each transaction.

Flow Records

AppFlow records contain standard NetFlow or IPFIX information, such as time stamps for the beginning and end of a flow, packet count, and byte count. AppFlow records also contain application-level information (such as HTTP URLs, HTTP request methods and response status codes, server response time, and latency). Webpage performance data (such as page load time, page render time, and time spent on the page). And database information (such as database protocol, database response status, and database response size). IPFIX flow records are based on templates that need to be sent before sending flow records.

Templates

AppFlow defines a set of templates, one for each type of flow. Each template contains a set of standard Information Elements (IEs) and Enterprise-specific Information Elements (EIEs). IPFIX templates define the order and sizes of the Information Elements (Internet Explorer) in the flow record. The templates are sent to the collectors at regular intervals, as described in RFC 5101.

A template can include the following EIEs:

- transactionID

  An unsigned 32-bit number identifying an application-level transaction. For HTTP, it corresponds to a request and response pair. All flow records that correspond to this request and response pair have the same transaction ID. In the most common case, there are four uniflow records that correspond to this transaction. If the NetScaler generates the response by itself (served from the integrated cache or by a security policy), there might be only two flow records for this transaction.

- connectionID

  An unsigned 32-bit number identifying a layer-4 connection (TCP or UDP). The NetScaler flows are bidirectional, with two separate flow records for each direction of the flow. This information element can be used to link the two flows.

For the NetScaler, connectionID is an identifier for the connection data structure to track the progress of a connection. In an HTTP transaction, for instance, a given connectionID might have multiple transactionID elements corresponding to multiple requests that were made on that connection.

- tcpRTT

  The round trip time, in milliseconds, as measured on the TCP connection. It can be used as a metric to determine the client or server latency on the network.

- httpRequestMethod

  An 8-bit number indicating the HTTP method used in the transaction. An options template with the number-to-method mapping is sent along with the template.

- httpRequestSize

  An unsigned 32-bit number indicating the request payload size.

- httpRequestURL

  The HTTP URL requested by the client.

- httpUserAgent

  The source of incoming requests to the Web server.

- httpResponseStatus

  An unsigned 32-bit number indicating the response status code.

- httpResponseSize

  An unsigned 32-bit number indicating the response size.

- httpResponseTimeToFirstByte

  An unsigned 32-bit number indicating the time taken to receive the first byte of the response.

- httpResponseTimeToLastByte

  An unsigned 32-bit number indicating the time taken to receive the last byte of the response.

- flowFlags

  An unsigned 64-bit flag used to indicate different flow conditions.

EIEs for webpage performance data

- clientInteractionStartTime

  Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and style sheets.

- clientInteractionEndTime

  Time at which the browser received the last byte of response to load all the objects of the page such as images, scripts, and style sheets.

- clientRenderStartTime

  Time at which the browser starts to render the page.

- clientRenderEndTime

  Time at which a browser finished rendering the entire page, including the embedded objects.

EIEs for database information

- dbProtocolName

  An unsigned 8-bit number indicating the database protocol. Valid values are 1 for MS SQL and 2 for MySQL.

- dbReqType

  An unsigned 8-bit number indicating the database request method used in the transaction. For MS SQL, valid values are 1 is for QUERY, 2 is for TRANSACTION, and 3 is for RPC. For valid values for MySQL, see the MySQL documentation.

- dbReqString

  Indicates the database request string without the header.

- dbRespStatus

  An unsigned 64-bit number indicating the status of the database response received from the web server.

- dbRespLength

  An unsigned 64-bit number indicating the response size.

- dbRespStatString

  The response status string received from the web server.

Configuring the AppFlow Feature

September 21, 2020

You configure AppFlow in the same manner as most other policy-based features. First, you enable the AppFlow feature. Then you specify the collectors to which the flow records are sent. After that, you define actions, which are sets of configured collectors. Then you configure one or more policies and associate an action to each policy. The policy tells the NetScaler appliance to select requests the flow records of which are sent to the associated action. Finally, you bind each policy either globally or to specific virtual server to put it into effect.

You can further set AppFlow parameters to specify the template refresh interval and to enable the exporting of httpURL, httpCookie, and httpReferer information. On each collector, you must specify the NetScaler IP address as the address of the exporter.

> **Note**
>
> For information about configuring the NetScaler as an exporter on the collector, see the documentation for the specific collector.

The configuration utility provides tools that help users define the policies and actions. It determines exactly how the NetScaler appliance export records for a particular flow to a set of collectors(action.) The command line interface provides a corresponding set of CLI-based commands for experienced users who prefer a command line.

Enabling AppFlow

To be able to use the AppFlow feature, you must first enable it.

> **Note**
>
> AppFlow can be enabled only on nCore NetScaler appliances.

To enable the AppFlow feature by using the command line interface

At the command prompt, type one of the following commands:

```
enable ns feature AppFlow
<!--NeedCopy-->
```

To enable the AppFlow feature by using the configuration utility

Navigate to **System > Settings**, click **Configure Advanced Features**, and select the **AppFlow** option.

Specifying a Collector

A collector receives AppFlow records generated by the NetScaler appliance. To send the AppFlow records, you must specify at least one collector. By default, the collector listens to IPFIX messages on UDP port 4739. You can change the default port, when configuring the collector. Similarly, by default, NSIP is used as the source IP for AppFlow traffic. You can change this default source IP to a SNIP address when configuring a collector. You can also remove unused collectors.

To specify a collector by using the command line interface

At the command prompt, type the following commands to add a collector and verify the configuration:

```
add appflow collector <name> -IPAddress <ipaddress> -port <port_number>
    -netprofile <netprofile_name>

show appflow collector <name>
<!--NeedCopy-->
```

Example

```
add appflow collector col1 -IPaddress 10.102.29.251 -port 8000 -
   netprofile n2
<!--NeedCopy-->
```

To specify multiple collectors by using the command line interface

At the command prompt, type the following commands to add and send same data to multiple collectors:

```
add appflow collector <collector1> -IPAddress <IP>

add appflow collector <collector2> -IPAddress <IP>

add appflow action <action> -collectors <collector1> <collector2>

add appflow policy <policy> true <action>

bind lbvserver <lbvserver> -policy <policy> -priority <priority>
<!--NeedCopy-->
```

To specify one or more collectors by using the configuration utility

Navigate to **System > AppFlow > Collectors**, and create the AppFlow collector.

Configuring an AppFlow Action

An AppFlow action is a set collector, to which the flow records are sent if the associated AppFlow policy matches.

To configure an AppFlow action by using the command line interface

At the command prompt, type the following commands to configure an AppFlow action and verify the configuration:

```
add appflow action <name> --collectors <string> ... [-
    clientSideMeasurements (Enabled|Disabled) ] [-comment <string>]

show appflow action
<!--NeedCopy-->
```

Example

```
add appflow action apfl-act-collector-1-and-3 -collectors collector-1
    collecter-3
<!--NeedCopy-->
```

To configure an AppFlow action by using the configuration utility

Navigate to **System > AppFlow > Actions**, and create the AppFlow action.

Configuring an AppFlow Policy

After you configure an AppFlow action, you must next configure an AppFlow policy. An AppFlow policy is based on a rule, which consists of one or more expressions.

> **Note**
>
> For creating and managing AppFlow policies, the configuration utility provides assistance that is not available at the command line interface.

To configure an AppFlow policy by using the command line interface

At the command prompt, type the following command to add an AppFlow policy and verify the configuration:

```
add appflow policy <name> <rule> <action>

show appflow policy <name>
<!--NeedCopy-->
```

Example

```
add appflow policy apfl-pol-tcp-dsprt client.TCP.DSTPORT.EQ(22) apfl-
    act-collector-1-and-3
<!--NeedCopy-->
```

To configure an AppFlow policy by using the configuration utility

Navigate to **System > AppFlow > Policies**, and create the AppFlow policy.

To add an expression by using the Add Expression dialog box

1. In the **Add Expression** dialog box, in the first list box choose the first term for your expression.
    - HTTP
      The HTTP protocol. Choose the if you want to examine some aspect of the request that pertains to the HTTP protocol.
    - SSL

      The protected websites. Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
    - CLIENT

      The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
      When you make your choice, the rightmost list box lists appropriate terms for the next part of your expression.
2. In the second list box, choose the second term for your expression. The choices depend upon which choice you made in the previous step, and are appropriate to the context. After you make your second choice, the Help window below the Construct Expression window (which was blank) displays help describing the purpose and use of the term you just chose.
3. Continue choosing terms from the list boxes that appear to the right of the previous list box, or typing strings or numbers in the text boxes that appear to prompt you to enter a value, until your expression is finished.

Binding an AppFlow Policy

To put a policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to the traffic related to that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

In the NetScaler operating system, policy priorities work in reverse order—the higher the number, the lower the priority. For example, if you have three policies with priorities of 10, 100, and 1000, the policy assigned a priority of 10 is performed first. Later, the policy assigned with a priority of 100, and finally the policy assigned an order of 1000.

You can leave yourself plenty of a room to add other policies in any order, and still set them to evaluate in the order you want. You can set priorities with intervals of 50 or 100 between each policy when you globally bind it. You can then add more policies at any time without having to change the priority of an existing policy.

To globally bind an AppFlow policy by using the command line interface

At the command prompt, type the following command to globally bind an AppFlow policy and verify the configuration:

```
bind appflow global <policyName> <priority> [<gotoPriorityExpression [-
    type <type>] [-invoke (<labelType> <labelName>)]

show appflow global
<!--NeedCopy-->
```

Example

```
bind appflow global af_policy_lb1_10.102.71.190 1 NEXT -type
    REQ_OVERRIDE -invoke vserver google
<!--NeedCopy-->
```

To bind an AppFlow policy to a specific virtual server by using the command line interface

At the command prompt, type the following command to bind an AppFlow policy to a specific virtual server and verify the configuration:

```
bind lb vserver <name> -policyname <policy_name> -priority <priority>
<!--NeedCopy-->
```

Example

```
bind lb vserver google -policyname af_policy_google_10.102.19.179 -priority 251
<!--NeedCopy-->
```

To globally bind an AppFlow policy by using the configuration utility

Navigate to **System > AppFlow**, click **AppFlow policy Manager**, and select the relevant Bind Point (Default Global) and Connection Type, and then bind the AppFlow policy.

To bind an AppFlow policy to a specific virtual server by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, select the virtual server, and click **Policies**, and bind the AppFlow policy.

Enabling AppFlow for Virtual Servers

If you want to monitor only the traffic through certain virtual servers, enable AppFlow specifically for those virtual servers. You can enable AppFlow for load balancing, content switching, cache redirection, SSL VPN, GSLB, and authentication virtual servers.

To enable AppFlow for a virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> <protocol> <IPAddress> <port> -appflowLog ENABLED
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 HTTP 10.102.29.161 80 -appflowLog ENABLED
<!--NeedCopy-->
```

To enable AppFlow for a virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select the virtual server, and enable **AppFlow Logging** option.

Enabling AppFlow for a Service

You can enable AppFlow for services that are to be bound to the load balancing virtual servers.

To enable AppFlow for a service by using the command line interface

At the command prompt, type:

```
set service <name> -appflowLog ENABLED
<!--NeedCopy-->
```

Example

```
set service ser -appflowLog ENABLED
<!--NeedCopy-->
```

To enable AppFlow for a service by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, select the service, and enable **AppFlow Logging** option.

Setting the AppFlow Parameters

You can set AppFlow parameters to customize the exporting of data to the collectors.

To set the AppFlow Parameters by using the command line interface

At the command prompt, type the following commands to set the AppFlow parameters and verify the settings:

```
set appflow param [-templateRefresh <secs>] [-appnameRefresh <secs>] [-
    flowRecordInterval <secs>] [-udpPmtu <positive_integer>] [-httpUrl (
     **ENABLED** | **DISABLED** )] [-httpCookie ( **ENABLED** | **
    DISABLED** )] [-httpReferer ( **ENABLED** | **DISABLED** )] [-
    httpMethod ( **ENABLED** | **DISABLED** )] [-httpHost ( **ENABLED**
    | **DISABLED** )] [-httpUserAgent ( **ENABLED** | **DISABLED** )] [-
    httpXForwardedFor ( **ENABLED** | **DISABLED** )][-clientTrafficOnly
     ( **YES** | **NO**)]

show appflow Param
<!--NeedCopy-->
```

Example

```
set appflow Param -templateRefresh 240 -udpPmtu 128 -httpUrl enabled
<!--NeedCopy-->
```

To set the AppFlow parameters by using the configuration utility

Navigate to **System > AppFlow**, click **Change AppFlow Settings**, and specify relevant AppFlow parameters.

Example: Configuring AppFlow for DataStream

The following example illustrates the procedure for configuring AppFlow for DataStream using the command line interface.

```
enable feature appflow

add db user sa password freebsd

add lbvserver lb0 MSSQL 10.102.147.97 1433 -appflowLog ENABLED

add service sv0 10.103.24.132 MSSQL 1433 -appflowLog ENABLED

bind lbvserver lb0 sv0

add appflow collector col0 -IPAddress 10.102.147.90

add appflow action act0 -collectors col0

add appflow policy pol0 "mssql.req.query.text.contains("select")" act0

bind lbvserver lb0 -policyName pol0 -priority 10
<!--NeedCopy-->
```

When the NetScaler appliance receives a database request, the appliance evaluates the request against a configured policy. If a match is found, the details are sent to the AppFlow collector configured in the policy.

Exporting Performance Data of Web Pages to AppFlow Collector

September 21, 2020

The EdgeSight Monitoring application provides web page monitoring data with which you can monitor the performance of various Web applications served in a NetScaler environment. You can now export this data to AppFlow collectors to get an in-depth analysis of the web page applications. AppFlow, which is based on IPFIX standard, provides more specific information about web application performance than does EdgeSight monitoring alone.

You can configure both load balancing and content switching virtual servers to export EdgeSight Monitoring data to AppFlow collectors. Before configuring a virtual server for AppFlow export, associate an Appflow action with the EdgeSight Monitoring responder policy.

The following web page performance data is exported to AppFlow:

- **Page Load Time**. Elapsed time, in milliseconds, from when the browser starts to receive the first byte of a response until the user starts to interact with the page. At this stage, all the page content might not be loaded.
- **Page Render Time**.Elapsed time, in milliseconds, from when the browser receives the first byte of response until either all page content has been rendered or the page load action has timed out.
- **Time Spent on the Page**. Time spent by users on a page. Represents the period of time from one page request to the next one.

AppFlow transmits the performance data by using the Internet Protocol Flow Information eXport (IPFIX) format, which is an open Internet Engineering Task Force (IETF) standard defined in RFC 5101. The AppFlow templates use the following enterprise-specific Information Elements (EIEs) to export the information:

- **Client Load End Time**. Time at which the browser received the last byte of a response to load all the objects of the page such as images, scripts, and stylesheets.
- **Client Load Start Time**.Time at which the browser receives the first byte of the response to load any objects of the page such as images, scripts, and stylesheets.
- **Client Render End Time**. Time at which browser finished rendering the entire page, including the embedded objects.
- **Client Render Start Time**. Time at which the browser started rendering the page.

Prerequisites for Exporting Performance Data of Web Pages to AppFlow Collectors

Before associating the AppFlow action with the AppFlow policy, verify that the following prerequisites have been met:

- The AppFlow feature has been enabled and configured.
- The Responder feature has been enabled.
- The EdgeSight Monitoring feature has been enabled.
- EdgeSight Monitoring has been enabled on the load balancing or content switching virtual servers bound to the services of applications for which you want to collect the performance

data.

Associating an AppFlow Action with the EdgeSight Monitoring Responder Policy

To export the web page performance data to the AppFlow collector, you must associate an AppFlow action with the EdgeSight Monitoring responder policy. An AppFlow action specifies which set of collectors receive the traffic.

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the command line interface

At the command prompt, type:

```
set responder policy <name> -appflowAction <action_Name>
<!--NeedCopy-->
```

Example

```
set responder policy pol -appflowAction actn
<!--NeedCopy-->
```

To associate an AppFlow action with the EdgeSight Monitoring Responder policy by using the configuration utility

1. Navigate to **AppExpert** > **Responder** > **Policies**.
2. In the details pane, select an EdgeSight Monitoring responder policy, and then click **Open**.
3. In the **Configure Responder Policy** dialog box, in the **AppFlow Action** drop-down list, select the AppFlow action associated with the collectors to which you want to send the web-page performance data.
4. Click **OK**.

Configuring a Virtual Server to Export EdgeSight Statistics to Appflow Collectors

To export EdgeSight statistics information from a virtual server to the AppFlow collector, you must associate an AppFlow action with the virtual server.

To associate an AppFlow action with a Load Balancing or Content Switching virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**. You can also navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**.
2. In the details pane, select a virtual server, or multiple virtual servers, and then click **Enable EdgeSight Monitoring**.
3. In the Enable EdgeSight Monitoring dialog box, select the **Export EdgeSight statistics to Appflow** check box.
4. From the Appflow Action drop-down list, select the **AppFlow** action. The AppFlow action defines the list of AppFlow collectors to which it exports EdgeSight Monitoring statistics. If you have selected multiple load balancing virtual servers, the same AppFlow Action will be associated with the responder policies bound to them. You can later change the AppFlow Action configured for each of the selected Load Balancing virtual server individually, if required.
5. Click **OK**.

Session reliability on NetScaler high availability pair

January 4, 2021

When a network disruption or a device failover occurs during an ICA session, session reconnection can use one of two mechanisms: Session Reliability or Auto Client Reconnect.

Session Reliability, the preferred mode, is a seamless experience for the user. The disruption is barely noticeable for brief network interruptions.

Auto Client Reconnect, the fallback option, involves restarting the client. This mechanism is disruptive for the user and is not always supported.

Receivers can now reconnect their ICA sessions seamlessly using the ICA Session Reliability feature, when HDX Insight is enabled.
This feature works both in standalone and in a NetScaler HA pair configuration and even when a NetScaler failover happens.

> **Note**
>
> - NetScaler appliances must be running on software version 11.1 build 49.16 or later.
> - Do not enable or disable Session Reliability mode when the NetScaler appliances have active connections.
> - Enabling or disabling the feature when connections are still active causes HDX Insight to stop parsing those sessions after a failover occurs and result in loss of information about the sessions.
> - Session Reliability on a high availability setup is disabled by default for NetScaler software version 11.1 49.16 or later. Session reliability is supported on a high availability setup only if

> both the nodes of the setup run the same build (for example, release 11.1 build 53). In other words, session reliability is not supported on a high availability setup if both the nodes run different builds (for example, one node has release 11.1 build 53 whereas the other has release 11.1 build 56).
>
> - When HDX Insight is enabled, basic encryption applications and desktops reconnect after high availability failover even if the `EnableSRonHAFailover parameter` is disabled.

To configure Session reliability from NetScaler CLI:

1. At the command line, use the default system administrator credentials to log on to the system.
2. To enable Session Reliability on HA failover, at the prompt, type: *set ica parameter EnableSRonHAFailover YES*
3. To disable Session Reliability on HA failover, at the prompt, type: *set ica parameter EnableSRonHAFailover NO*

To Enable Session reliability on HA failover from NetScaler GUI:

1. In a web browser, type the IP address of the primary NetScaler instance in the HA pair (for example, http://192.168.100.1).
2. In **User Name** and **Password**, enter the administrator credentials.
3. On the **Configuration** tab, navigate to **System** > **Settings**, and click **Change ICA Parameters**.
4. In the **Change ICA Parameters** section, select **Session Reliability on HA Failover**.
5. Click **OK**.

Limitations

- Enabling this feature results in increased bandwidth consumption which is due to ICA compression being turned off by the feature, and the extra traffic between the primary and secondary nodes to keep them in sync.
- This feature is supported in Active-Passive mode only. Active-Active mode is not currently supported.
- When HDX Insight is enabled, and Session Reliability on the HA knob is set to NO, only ACR reconnect mode is supported in the NetScaler high availability failover scenario. The HA knob does not disable Session Reliability if HDX Insight is disabled.

The **Session Reconnect Semantics** table is as follows:

Session Reconnect Semantics

Status	EnableSRonHAFailover Yes	EnableSRonHAFailover No (Default)
HDX Insight enabled	Session reconnect for ICA Sessions works	Session reconnect for ICA Sessions does not work
HDX Insight disabled	Session reconnect for ICA Sessions works	Session reconnect for ICA Sessions works

> **Points to note**

- Session Reliability for ICA sessions works out of the box with Gateway.
- Session Reliability for ICA sessions does not work ONLY when both the following conditions are met:
  - HDX Insight is enabled
  - EnableSRonHAFailover is set to NO
- Setting the `EnableSRonHAFailover knob` to either YES or NO does not make any difference, when HDX Insight is disabled.

Application Firewall

September 21, 2020

The following topics cover installation and configuration of the Citrix Application Firewall feature.

Introduction	An overview of web security and how the Web App Firewall works.
Configuration	How to configure the Web App Firewall to protect a web site, a web service, or a Web 2.0 site.
Signatures	A detailed description of the signatures feature and how to configure the signatures, add signatures from a supported vulnerability scanning tool, and define your own signatures, with examples.
Overview of Security Checks	A detailed description of all of the Web App Firewall security checks, with configuration information and examples.

Profiles	A description of how profiles are configured and used in the Web App Firewall.
Policies	A description of how policies are used when configuring the Web App Firewall, with examples of useful policies.
Imports	A description of how the Web App Firewall uses different types of imported files, and how to import and export files.
Global Configuration	A description of Web App Firewall features that apply to all profiles, and how to configure them.
Use Cases	Extended examples that demonstrate how to set up the Web App Firewall to best protect specific types of more complex web sites and web services.
Logs, Statistics, and Reports	How to access and use the Web App Firewall logs, the statistics, and the reports to assist in configuring the Web App Firewall.

The Citrix Application Firewall offers easy to configure options to meet a wide range of application security requirements. Web App Firewall profiles, which consist of sets of security checks, can be used to protect both the requests and the responses by providing deep packet-level inspections. Each profile includes an option to select basic protections or advanced protections. Some protections might require use of other files. For example, xml validation checks might require WSDL or schema files. The profiles can also use other files, such as signatures or error objects. These files can be added locally, or they can be imported ahead of time and saved on the appliance for future use. They can be shared by multiple profiles.

Profiles work in conjunction with the Web App Firewall policies. Each policy identifies a type of traffic, and that traffic is inspected for the security check violations specified in the profile that is associated with the policy. The policies can have different bind points, which determine the scope of the policy. For example, a policy that is bound to a specific virtual server is invoked and evaluated for only the traffic flowing through that virtual server. The policies are evaluated in the order of their designated priorities, and the first one that matches the request or response is applied.

- Quick Deployment of Web App Firewall Protection

  You can use the following procedure for quick deployment of Web App Firewall security:

1. Add an appfw profile and select the appropriate type (html, xml, JSON) for the security requirements of the application.
2. Select the required level of security (basic or advanced).
3. Add or import the required files, such as signatures or WSDL.
4. Configure the profile to use the files, and make any other necessary changes to the default settings.
5. Add an appfw policy for this profile.
6. Bind the policy to the target bind point and specify the priority.

- Web App Firewall entities

  Following are brief overviews of the Web App Firewall entities. For details, see the Web App Firewall Guide.

  Profile—An Web App Firewall profile specifies what to look for and what to do. It inspects both the request and the response to determine which potential security violations should be checked and what actions should be taken when processing a transaction. A profile can protect an HTML, XML or HTML and XML payload. Depending on the security requirements of the application, you can create either a basic or an advanced profile. A basic profile can protect against known attacks. If higher security is required, you can deploy an advanced profile to allow controlled access to the application resources, blocking zero day attacks. However, a basic profile can be modified to offer advanced protections, and vice versa. Multiple action choices (for example, block, log, learn, and transform) are available. Advanced security checks might use session cookies and hidden form tags for controlling and monitoring the client connections. Web App Firewall profiles can learn the triggered violations and suggest the relaxation rules.

  Basic Protections—A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests should be allowed and which should be denied. Incoming requests are matched against these lists and the configured actions are applied. This allows the user to be able to secure applications with minimal configuration for relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-site scripting can also be easily detected.

  Advanced Protections—As the name indicates, advanced protections are used for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or cross-site request forgery attacks. Learning, which observes the traffic and deploys the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing and do not allow use of caching, which can affect performance.

Import—Import functionality is useful when Web App Firewall profiles need to use external files, that is, files hosted on an external or internal web server, or that have to be copied from a local machine. Importing a file and storing it on the appliance is very useful, especially in situations where you have to control access to external websites, or where compilation takes a long time, large files have to be synced across HA deployments, or you can reuse a file by copying it across multiple devices. For example:

- <WSDLs hosted on external web servers can be imported locally before blocking access to external websites.
- <Large signature files generated by an external scan tool such as Cenzic can be imported and precompiled, using schema on the Citrix appliance.
- <A customized HTML or XML error page can be imported from an external web server or copied from a local file.

Signatures—Signatures are very powerful, because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple choices (for example, block, log, learn, and transform) are available for the action to take when a signature match is detected. The Web App Firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the auto-update feature. Rules created by other scan tools can also be imported. The signature object can be customized by adding new rules, which can work in conjunction with the other security checks specified in the Web App Firewall profile. A signature rule can have multiple patterns and can flag a violation only when all the patterns are matched, thereby avoiding false positives. Careful selection of a literal fastmatch pattern for a rule can significantly optimize processing time.

Policies—Web App Firewall Policies are used to filter and separate the traffic into different types. This provides the flexibility to implement different levels of security protections for the application data. Access to highly sensitive data can be directed to advanced security-check inspections, while less sensitive data is protected by basic-level security inspections. Policies can also be configured to bypass security-check inspection for harmless traffic. Higher security requires more processing, so careful design of the policies can provide desired security along with optimized performance. The priority of the policy determines the order in which it is evaluated, and its bind point determines the scope of its application.

Highlights

1. <Ability to secure a wide range of applications by protecting different types of data, implementing the right level of security for different resources, and still getting maximum performance.

2. Flexibility to add or modify a security configuration. You can tighten or relax security checks by enabling or disabling basic and advanced protections.
3. Option to convert an HTML profile to an XML or Web2.0 (HTML+XML) profile and vice-versa, providing the flexibility to add security for different types of payload.
4. Easily deployed actions to block attacks, monitor them in logs, collect statistics, or even transform some attack strings to render them harmless.
5. Ability to detect attacks by inspecting incoming requests, and to prevent leakage of sensitive data by inspecting the responses sent by the servers.
6. Capability to learn from the traffic pattern to get recommendations for easily editable relaxation rules that can be deployed to allow exceptions.
7. Hybrid security model that applies the power of customizable signatures to block attacks that match specified patterns, and provides the flexibility to use the positive-security-model checks for basic or advanced security protections.
8. Availability of comprehensive configuration reports, including information about PCI-DSS compliance.

FAQs and Deployment Guide

September 21, 2020

Q: Why is NetScaler App Firewall the preferred choice for securing applications?

**With the following features, the NetScaler App App Firewall offers a comprehensive security solution:

- **Hybrid security model:** NetScaler hybrid security model allows you to take advantage of both a positive security model and a negative security model to come up with a configuration ideally suited for your applications.
  - **Positive security model** protects against Buffer Overflow, CGI-BIN Parameter Manipulation, Form/Hidden Field Manipulation, Forceful Browsing, Cookie or Session Poisoning, Broken ACLs, Cross-Site Scripting (XSS), Command Injection, SQL Injection, Error Triggering Sensitive Information Leak, Insecure Use of Cryptography, Server Misconfiguration, Back Doors and Debug Options, Rate-Based Policy Enforcement, Well Known Platform Vulnerabilities, Zero-Day Exploits, Cross Site Request Forgery (CSRF), and leakage of Credit Card and other sensitive data.
  - **Negative security model** uses a rich set signatures to protect against L7 and HTTP application vulnerabilities. The App Firewall is integrated with several third party scanning tools, such as those offered by Cenzic, Qualys, Whitehat, and IBM. The built-in XSLT files

allow easy importation of rules, which can be used in conjunction with the native-format Snort based rules. An auto-update feature gets the latest updates for new vulnerabilities.

The positive security model might be the preferred choice for protecting applications that have a high need for security, because it gives you the option to fully control who can access what data. You allow only what you want and block the rest. This model includes a built-in security check configuration, which is deployable with few clicks. However, keep in mind that the tighter the security, the greater the processing overhead.

The negative security model might be preferable for customized applications. The signatures allow you to combine multiple conditions, and a match and the specified action are triggered only when all the conditions are satisfied. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions.

- **Request as well as response side detection and protection:** You can inspect the incoming requests to detect any suspicious behavior and take appropriate actions, and you can check the responses to detect and protect against leakage of sensitive data.
- **Rich set of built-in protections for HTML, XML and JSON payloads:** The App Firewall offers 19 different security checks. Six of them (such as Start URL and Deny URL) apply to both HTML and XML data. Five checks (such as Field Consistency and Field Format) are specific to HTML, and eight (such as XML Format and Web Service Interoperability) are specific to XML payloads. This feature includes a rich set of actions and options. For example, URL Closure enables you to control and optimize the navigation through your website, to safeguard against forceful browsing without having to configure relaxation rules to allow each and every legitimate URL. You have the option to remove or x-out the sensitive data, such as credit-card numbers, in the response. Be it SOAP Array attack protection, XML denial of Service (XDoS), WSDL scan prevention, Attachment check, or any number of other XML attacks, you have the comfort of knowing that you have an ironclad shield protecting your data when your applications are protected by the App Firewall. The signatures allow you to configure rules using XPATH-Expressions to detect violations in the body as well as the header of a JSON payload.
- **GWT:** Support for protecting Google Web Toolkit applications to safeguard against SQL, XSS and Form Field Consistency check violations.
- **Java-free, user friendly graphical user interface (GUI):** An intuitive GUI and preconfigured security checks make it easy to deploy security by clicking a few buttons. A wizard prompts and guides you to create the required elements, such as profiles, policies, signatures, and bindings. The HTML5 based GUI is free of any Java dependency. It's performance is significantly better than that of the older, Java based versions.
- **Easy to Use and automatable CLI:** Most of the configuration options that are available in GUI

are also available in the command line interface (CLI). The CLI commands can be executed by a batch file and are easy to automate.

- **Support for REST API:** The NetScaler NITRO protocol supports a rich set of REST API's to automate App Firewall configuration and collect pertinent statistics for ongoing monitoring of security violations.
- **Learning:** The App Firewall's ability to learn by monitoring traffic to fine tune security is very user friendly. The learning engine recommends rules, which makes it easy to deploy relaxations without proficiency in regular expressions.
- **RegEx editor support:** Regular expression offer an elegant solution to the dilemma of wanting to consolidate rules and yet optimize search. You can capitalize on the power of regular expressions to configure URLs, field names, signature patterns, and so on. The rich built-in GUI RegEx editor offers you a quick reference for the expressions and provides a convenient way to validate and test your RegEx for accuracy.
- **Customized error page:** Blocked requests can be redirected to an error URL. You also have the option to display a customized error object that uses supported variables and Citrix default syntax (advanced PI expressions) to embed troubleshooting information for the client.
- **PCI-DSS, stats, and other violation reports:** The rich set of reports makes it easy to meet the PCI-DSS compliance requirement, gather stats about traffic counters, and view violation reports for all profiles or just one profile.
- **Logging and click-to-rule from log:** Detailed logging is supported for native as well as CEF format. The App Firewall offers you the ability to filter targeted log messages in the syslog viewer. You can select a log message and deploy a corresponding relaxation rule by a simple click of a button. You have the flexibility to customize log messages and also have support for generating web logs. For additional details, see App Firewall Logstopic.
- **Include violation logs in trace records:** The ability to include log messages in the trace records makes it very easy to debug unexpected behavior such as reset and block.
- **Cloning:** The useful Import/Export profile option allows you to clone the security configuration from one NetScaler appliance to others. Export learned data options make it easy to export the learned rules to an Excel file. You can then get them reviewed and approved by application owner before applying them.
- **An AppExpert template** (a set of configuration settings) can be designed to provide appropriate protection for your websites. You can simplify and expedite the process of deploying similar protection on other appliances by exporting these cookie-cutter templates to a template. For additional details, see AppExpert Template
- **Sessionless security checks:** Deploying sessionless security checks can help you reduce the memory footprint and expedite the processing.

- **Interoperability with other NetScaler features:** The App Firewall works seamlessly with other NetScaler features, such as rewrite, URL transformation, integrated caching, CVPN, and rate limiting.
- **Support of PI expressions in policies:** You can leverage the power of advanced PI expressions to design policies to implement different levels of security for different parts of your application.
- **Support for IPv6:** The App Firewall supports both IPv4 and IPv6 protocols.
- **Geolocation based security protection:** You have the flexibility of using Citrix default syntax (PI Expressions) for configuring location based policies, which can be used in conjunction with a built-in location database to customize firewall protection. You can identify the locations from which malicious requests originate, and enforce the desired level of security-check inspections for requests that originate from a specific geographical location.
- **Performance:** Request-side **streaming** significantly improves performance. As soon as a field is processed, the resulting data is forwarded to the back end while evaluation continues for the remaining fields. The improvement in processing time is especially significant when handling large posts.
- **Other security features:** The App Firewall has several other security knobs that can help ensure the security of your data. For example, the **Confidential Field** lets you block leakage of sensitive information in the log messages, and **Strip HTML Comment** allows you to remove the HTML comments from the response before forwarding it to the client. **Field Types** can be used to specify what inputs are allowed in the forms submitted to your application.

Q: What do I need to do to configure App Firewall?

Do the following:

- Add an App Firewall profile and select the appropriate type (html, xml, web2.0) for the security requirements of the application.
- Select the required level of security (basic or advanced).
- Add or import the required files, such as signatures or WSDL.
- Configure the profile to use the files, and make any other necessary changes to the default settings.
- Add an App Firewall policy for this profile.
- Bind the policy to the target bind point and specify the priority.

Q: How do I know what profile type to choose?

The App Firewall profile offers protection for both HTML and XML payloads. Depending on the need of your application, you can choose either a HTML profile or XML profile. If your application supports both HTML and XML data, you can choose a Web2.0 profile.

Q: What is the difference between basic and advanced profiles? How do I decide which one I need?

The decision to use a basic or an advance profile depends on the security need of your application. A basic profile includes a preconfigured set of Start URL and Deny URL relaxation rules. These relaxation rules determine which requests are allowed and which are denied. Incoming requests are matched with the preconfigured rules, and the configured actions are applied. The user can secure applications with minimal configuration of relaxation rules. The Start URL rules protect against forceful browsing. Known web server vulnerabilities that are exploited by hackers can be detected and blocked by enabling a set of default Deny URL rules. Commonly launched attacks, such as Buffer Overflow, SQL, or Cross-Site Scripting can also be easily detected.

As the name indicates, advanced protections are for applications that have higher security requirements. Relaxation rules are configured to allow access to only specific data and block the rest. This positive security model mitigates unknown attacks, which might not be detected by basic security checks. In addition to all the basic protections, an advanced profile keeps track of a user session by controlling the browsing, checking for cookies, specifying input requirements for various form fields, and protecting against tampering of forms or Cross-Site Request Forgery attacks. Learning, which observes the traffic and recommends the appropriate relaxations, is enabled by default for many security checks. Although easy to use, advanced protections require due consideration, because they offer tighter security but also require more processing. Some advance security checks do not allow use of caching, which can affect performance.

Keep the following points in mind when deciding whether to use basic or advanced profiles:

- Basic and advanced profiles are just starting templates. You can always modify the basic profile to deploy advanced security features, and vice versa.
- Advanced security checks require more processing and can affect performance. Unless your application needs advanced security, you might want to start with a basic profile and tighten the security as required for your application.
- You do not want to enable all security checks unless your application needs it.

Q: What is a policy? How do I select the bind point and set the priority?

App Firewall policies can help you sort your traffic into logical groups for configuring different levels of security implementation. Carefully select the bind points for the policies to determine which traffic is matched against which policy. For example, if you want every incoming request to be checked for SQL/XSS attacks, you can create a generic policy and bind it globally. Or, if you want to apply more stringent security checks to the traffic of a virtual server hosting applications that contain sensitive data, you can bind a policy to that virtual server.

Careful assignment of priorities can enhance the traffic processing. You want to assign higher priorities to more specific policies and lower priorities to generic policies. Note that the higher the number, the lower the priority. A policy with a priority of 10 is evaluated before a policy that has a priority of 15.

You can apply different levels of security for different kinds of contents, e.g. requests for static objects like images and text can be by-passed by using one policy and requests for other sensitive contents can be subjected to a much stringent check by using a second policy.

Q: How do I go about configuring the rules to secure my application?

The App Firewall makes it very easy to design the right level of security for your web-site. You can have multiple App Firewall policies, bound to different App Firewall profiles, to implement different levels of security-check inspections for your applications. You can initially monitor the logs to observe what security threats are being detected and which violations are being triggered. You can either manually add the relaxation rules or take advantage of the App Firewall's recommended learned rules to deploy the required relaxations to avoid false positives.

The NetScaler App Firewall offers **visualizer** support in GUI, which makes rule management very easy. You can easily view all the data on one screen, and take action on several rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate several rules. You can select a subset of the rules, basing your selection on the delimiter and Action URL. Visualizer support is available for viewing 1) learned rules and 2) relaxation rules.

1) The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. You can also skip (ignore) rules.

2) The visualizer for deployed relaxations offers you the option to add a new rule or edit an existing one. You can also enable or disable a group of rules by selecting a node and clicking the **Enable** or **Disable** button in the relaxation visualizer.

Q: What are signatures? How do I know which signatures to use?

A signature is an object that can have multiple rules. Each rule consists of one or more patterns that can be associated with a specified set of actions. The App Firewall has a built-in default signature object consisting of more than 1,300 signature rules, with an option to get the latest rules by using the **auto-update** feature to get protection against new vulnerabilities. Rules created by other scan tools can also be imported.

Signatures are very powerful because they use pattern matching to detect malicious attacks and can be configured to check both the request and the response of a transaction. They are a preferred option when a customizable security solution is needed. Multiple action choices (for example, block, log, learn, and transform) are available for when a signature match is detected. The default signatures cover rules to protect different types of applications, such as web-cgi, web-coldfusion, web-frontpage, web-iis, web-php, web-client, web-activex, web-shell-shock, and web-struts. To match the needs of your application, you can select and deploy the rules belonging to a specific category.

Signature-usage tips:

- You can just make a copy of the default signature object and modify it to enable the rules you need and configure the actions you want.
- The signature object can be customized by adding new rules, which can work in conjunction with other signature rules.
- The signature rules can also be configured to work in conjunction with the security checks specified in the App Firewall profile. If a match indicating a violation is detected by a signature as well as a security check, the more restrictive action is the one that gets enforced.
- A signature rule can have multiple patterns and be configured to flag a violation only when all the patterns are matched, thereby avoiding false positives.
- Careful selection of a literal fast-match pattern for a rule can significantly optimize processing time.

Q: Does the App Firewall work with other NetScaler features?

The App Firewall is fully integrated into the NetScaler appliance and works seamlessly with other features. You can configure maximum security for your application by using other NetScaler security features in conjunction with the App Firewall. For example, **AAA-TM** can be used to authenticate the user, check the user's authorization to access the content, and log the accesses, including invalid login attempts. **Rewrite** can be used to modify the URL or to add, modify or delete headers, and **Responder** can be used to deliver customized content to different users. You can define the maximum load for your website by using **Rate Limiting** to monitor the traffic and throttle the rate if it is too high. **HTTP Denial-of-Service (DoS)** protection can help distinguish between real HTTP clients and malicious DoS clients. You can narrow the scope of security-check inspection by binding the App Firewall policies to virtual servers, while still optimizing the user experience by using the **Load Balancing** feature to manage heavily used applications. Requests for static objects such as images or text can bypass security check inspection, taking advantage of **integrated caching** or **compression** to optimize the bandwidth usage for such content.

Q: How is the payload processed by the App Firewall and the other NetScaler features?

A diagram showing details of the L7 packet flow in a NetScaler appliance is available in the Processing Order of Features section at Getting Started topic.

Q: What is the recommended workflow for App Firewall deployment?

Now that you know the advantages of using the state-of-the-art security protections of the NetScaler App Firewall, you might want to collect additional information that can help you design the optimal solution for your security needs. Citrix recommends that you do the following:

- **Know your environment:** Knowing your environment will help you to identify the best security protection solution (signatures, security checks, or both) for your needs. Before you begin configuration, you should gather the following information.
  - **OS:** What kind of OS (MS Windows, Linux, BSD, Unix, others) do you have?
  - **Web Sever:** What web server (IIS, Apache or NetScaler Enterprise Server) are you running?
  - **Application:** What type of applications are running on your application server (for example, ASP.NET, PHP, Cold Fusion, ActiveX, FrontPage, Struts, CGI, Apache Tomcat, Domino, and WebLogic)?
  - Do you have customized applications or off-the-shelf (for example, Oracle, SAP) applications? What version you are using?
  - **SSL:** Do you require SSL? If so, what key size (512, 1024, 2048, 4096) is used for signing certificates?
  - **Traffic Volume:** What is the average traffic rate through your applications? Do you have seasonal or time-specific spikes in the traffic?
  - **Server Farm:** How many servers do you have? Do you need to use load balancing?
  - **Database:** What type of database (MS-SQL, MySQL, Oracle, Postgres, SQLite, nosql, Sybase, Informix etc.) do you use?
  - **DB Connectivity:** What kind of data base connectivity do you have (DSN, per-file connection string, single file connection string) and what drivers are used?
- **Identify your security needs:** You might want to evaluate which applications or specific data need maximum security protection, which ones are less vulnerable, and the ones for which security inspection can safely be bypassed. This will help you in coming up with an optimal configuration, and in designing appropriate policies and bind points to segregate the traffic. For example, you might want to configure a policy to bypass security inspection of requests for static web content, such as images, MP3 files, and movies, and configure another policy to apply advanced security checks to requests for dynamic content. You can use multiple policies and profiles to protect different contents of the same application.
- **License requirement:** Citrix offers a unified solution to optimize the performance of your application by taking advantage of a rich set of features such as load balancing, content switching, caching, compression, responder, rewrite, and content filtering, to name a few. Identifying the features that you want can help you decide which license you need.
- **Install and baseline a NetScaler appliance:** Create a virtual server and run test traffic through it to get an idea of the rate and amount of traffic flowing through your system. This information will help you to identify your capacity requirement and select the right appliance (VPX, MPX, or SDX).

- **Deploy the App Firewall:** Use the App Firewall wizard to proceed with a simple security configuration. The wizard walks you through several screens and prompts you to add a profile, policy, signature, and security checks.
  - **Profile:** Select a meaningful name and the appropriate type (HTML, XML or WEB 2.0) for your profile. The policy and signatures will be auto-generated using the same name.
  - **Policy:** The auto-generated policy has the default Expression (true), which selects all traffic and is bound globally. This is a good starting point unless you have in mind a specific policy that you want to use.
  - **Protections:** The wizard helps you take advantage of the hybrid security model, in which you can use the default signatures offering a rich set of rules to protect different types of applications. **Simple** edit mode allows you to view the various categories (CGI, Cold Fusion, PHP, etc.). You can select one or more categories to identify a specific set of rules applicable to your application. Use the **Action** option to enable all the signature rules in the selected categories. Make sure that blocking is disabled, so that you can monitor the traffic before tightening the security. Click **Continue**. In the **Specify Deep protections** pane, you can make changes as needed to deploy the security check protections. In most cases, basic protections are sufficient for initial security configuration. Run the traffic for a while to collect a representative sample of the security-inspection data.
  - **Tightening the security:** After deploying App Firewall and observing the traffic for a while, you can start tightening the security of your applications by deploying relaxations and then enabling blocking. **Learning**, **Visualizer**, and **Click to deploy rules** are useful features that make it very easy to tweak your configuration to come up with just the right level of relaxation. At this point, you can also change the policy expression and/or configure additional policies and profiles to implement desired levels of security for different types of content.
  - **Debugging:** If you see unexpected behavior of your application, the App Firewall offers various options for easy debugging:
    - **Log**. If legitimate requests are getting blocked, your first step is to check the ns.log file to see if any unexpected security-check violation is being triggered.
    - **Disable feature.** If you do not see any violations but are still seeing unexpected behavior, such as an application resetting or sending partial responses, you can disable the App Firewall feature for debugging. If the issue persists, it rules out the App Firewall as a suspect.
    - **Trace records with log messages.** If the issue appears to be App Firewall related and needs closer inspection, you have the option to include security violation messages in an nstrace. You can use "Follow TCP stream" in the trace to view the details of the individual transaction, including headers, payload, and the corresponding log message, together on the same screen. Details of how to use this functionality are available at Appendixes.

Introduction

September 21, 2020

The NetScaler App App Firewall prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information. It does so by filtering both requests and responses, examining them for evidence of malicious activity, and blocking those that exhibit such activity. Your site is protected not only from common types of attacks, but also from new, as yet unknown attacks. In addition to protecting web servers and web sites from unauthorized access and misuse by hackers and malicious programs, the App Firewall provides protection against security vulnerabilities in legacy CGI code or scripts, other web frameworks, web server software, and the underlying operating systems.

The NetScaler App Firewall is available as a stand-alone appliance, or as a feature on a NetScaler virtual appliance (VPX). In the App Firewall documentation, the term NetScaler refers to the platform on which the App Firewall is running, regardless of whether that platform is a dedicated firewall appliance, a NetScaler on which other features have also been configured, or a NetScaler VPX.

To use the App Firewall, you must create at least one security configuration to block connections that violate the rules that you set for your protected web sites. The number of security configurations that you might want to create depends on the complexity of your web site. In some cases, a single configuration is sufficient. In other cases, particularly those that include interactive web sites, web sites that access database servers, online stores with shopping carts, you might need several different configurations to best protect sensitive data without wasting significant effort on content that is not vulnerable to certain types of attacks. You can often leave the defaults for the global settings, which affect all security configurations, unchanged. However, you can change the global settings if they conflict with other parts of your configuration or you prefer to customize them.

Web application security

September 21, 2020

Web application security is network security for computers and programs that communicate by using the HTTP and HTTPS protocols. This is an extremely broad area in which security flaws and weaknesses abound. Operating systems on both servers and clients have security issues and are vulnerable to attack. Web server software and web site enabling technologies such as CGI, Java, JavaScript, PERL and PHP have underlying vulnerabilities. Browsers and other client applications that communicate with web-enabled applications also have vulnerabilities. Web sites that use any technology but the simplest of HTML, including any site that allows interaction with visitors, often have vulnerabilities of their own.

In the past, a breach in security was often just an annoyance, but today that is seldom the case. For example, attacks in which a hacker gained access to a web server and made unauthorized modifications to (defaced) a web site used to be common. They were usually launched by hackers who had no motivation beyond demonstrating their skills to fellow hackers or embarrassing the targeted person or company. Most current security breaches, however, are motivated by a desire for money. The majority attempt to accomplish one or both of the following goals: to obtain sensitive and potentially valuable private information, or to obtain unauthorized access to and control of a web site or web server.

Certain forms of web attacks focus on obtaining private information. These attacks are often possible even against web sites that are secure enough to prevent an attacker from taking full control. The information that an attacker can obtain from a web site can include customer names, addresses, phone numbers, social security numbers, credit card numbers, medical records, and other private information. The attacker can then use this information or sell it to others. Much of the information obtained by such attacks is protected by law, and all of it by custom and expectation. A breach of this type can have extremely serious consequences for customers whose private information is compromised. At best, these customers will have to exercise vigilance to prevent others from abusing their credit cards, opening unauthorized credit accounts in their name, or appropriating their identities outright (identity theft). At worst, the customers may face ruined credit ratings or even be blamed for criminal activities in which they had no part.

Other web attacks are aimed at obtaining control of (or *compromising*) a web site or the server on which it operates, or both. A hacker who gains control of a web site or server can use it to host unauthorized content, act as a proxy for content hosted on another web server, provide SMTP services to send unsolicited bulk email, or provide DNS services to support such activities on other compromised web servers. Most web sites that are hosted on compromised web servers promote questionable or outright fraudulent businesses. For example, the majority of phishing web sites and child exploitation web sites are hosted on compromised web servers.

Protecting your web sites and web services against these attacks requires a multilayered defense capable of both blocking known attacks with identifiable characteristics and protecting against unknown attacks, which can often be detected because they look different from the normal traffic to your web sites and web services.

Known web attacks

The first line of defense for your web sites is protection against the large number of attacks that are known to exist and have been observed and analyzed by web security experts. Common types of attacks against HTML-based web sites include:

- **Buffer overflow attacks.** Sending an extremely long URL, extremely long cookie, or other extremely long bit of information to a web server in hopes of causing it or the underlying operating system to hang, crash, or provide the attacker with access to the underlying operating system. A buffer overflow attack can be used to gain access to unauthorized information, to compromise a web server, or both.

- **Cookie security attacks.** Sending a modified cookie to a web server, usually in hopes of obtaining access to unauthorized content by using falsified credentials.
- **Forceful browsing.** Accessing URLs on a web site directly, without navigating to the URLs by means of hyperlinks on the home page or other common start URLs on the web site. Individual instances of forceful browsing may simply indicate a user who bookmarked a page on your web site, but repeated attempts to access nonexistent content, or content that users should never access directly, often represent an attack on web site security. Forceful browsing is normally used to gain access to unauthorized information, but can also be combined with a buffer overflow attack in an attempt to compromise your server.
- **Web form security attacks.** Sending inappropriate content to your web site in a web form. Inappropriate content can include modified hidden fields, HTML or code in a field intended for alphanumeric data only, an overly long string in a field that accepts only a short string, an alphanumeric string in a field that accepts only an integer, and a wide variety of other data that your web site does not expect to receive in that web form. A web form security attack can be used either to obtain unauthorized information from your web site or to compromise the web site outright, usually when combined with a buffer overflow attack.

Two specialized types of attacks on web form security deserve special mention:

- **SQL injection attacks.** Sending an active SQL command or commands in a web form or as part of a URL, with the goal of causing an SQL database to execute the command or commands. SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a URL or a script on a web page to violate the same-origin policy, which forbids any script from obtaining properties from or modifying any content on a different web site. Since scripts can obtain information and modify files on your web site, allowing a script access to content on a different web site can provide an attacker the means to obtain unauthorized information, to compromise a web server, or both.

Attacks against XML-based web services normally fall into at least one of the following two categories: attempts to send inappropriate content to a web service, or attempts to breach security on a web service. Common types of attacks against XML-based web services include:

- **Malicious code or objects.** XML requests that contain code or objects that can either directly obtain sensitive information or can give an attacker control of the web service or underlying server.
- **Badly-formed XML requests.** XML requests that do not conform to the W3C XML specification, and that can therefore breach security on an insecure web service
- **Denial of service (DoS) attacks.** XML requests that are sent repeatedly and in high volume, with the intent of overwhelming the targeted web service and denying legitimate users access

to the web service.

In addition to standard XML-based attacks, XML web services and Web 2.0 sites are also vulnerable to SQL injection and cross-site scripting attacks, as described below:

- **SQL injection attacks.** Sending an active SQL command or commands in an XML-based request, with the goal of causing an SQL database to execute that command or commands. As with HTML SQL injection attacks, XML SQL injection attacks are normally used to obtain unauthorized information.
- **Cross-site scripting attacks.** Using a script included in an XML based application to violate the same-origin policy, which does not allow any script to obtain properties from or modify any content on a different application. Since scripts can obtain information and modify files by using your XML application, allowing a script access to content belonging to a different application can give an attacker the means to obtain unauthorized information, to compromise the application, or both

Known web attacks can usually be stopped by filtering web site traffic for specific characteristics (signatures) that always appear for a specific attack and should never appear in legitimate traffic. This approach has the advantages of requiring relatively few resources and posing relatively little risk of false positives. It is therefore a valuable tool in fighting attacks on web sites and web services, and configuring basic signature protections that intercept most known web attacks is easy to do.

Unknown web attacks

The greatest threat against web sites and applications does not come from known attacks, but from unknown attacks. Most unknown attacks fall into one of two categories: newly-launched attacks for which security firms have not yet developed an effective defense (zero-day attacks), and carefully-targeted attacks on a specific web site or web service rather than many web sites or web services (spear attacks). These attacks, like known attacks, are usually intended to obtain sensitive private information, compromise the web site or web service and allow it to be used for further attacks, or both of those goals.

Zero-day attacks are a major threat to all users. These attacks are usually of the same types as known attacks; zero-day attacks often involve injected SQL, a cross-site script, a cross-site request forgery, or another type of attack similar to known attacks. In most cases, they target vulnerabilities that the developers of the targeted software, web site, or web service either are unaware of or have just learned about. Security firms have therefore usually not developed defenses against these attacks, and even if they have, users have usually not obtained and installed the patches or performed the workarounds necessary to protect against these attacks. The time between discovery of a zero-day attack and availability of a defense (the vulnerability window) is shrinking, but perpetrators can still count on hours or even days in which many web sites and web services lack any specific protection against the attack.

Spear attacks are a major threat, but to a more select group of users. A common type of spear attack, a spear phish, is usually targeted at customers of a specific bank or financial institution, or (less commonly) at employees of a specific company or organization. Unlike other phishes, which are often crudely written forgeries that a user with any familiarity with the actual communications of that bank or financial institution can recognize, spear phishes are letter perfect and extremely convincing. They can contain information specific to the individual that, at first look, no stranger should know or be able to obtain. The spear phisher is therefore able to convince his or her target to provide the requested information, which the phisher can then use to loot accounts, to process illegitimately obtained money from other sources, or to gain access to other, even more sensitive information.

Both of these types of attack have certain characteristics that can usually be detected, although not by using static patterns that look for specific characteristics, as do standard signatures. Detecting these types of attacks requires more sophisticated and more resource-intensive approaches, such as heuristic filtering and positive security model systems. Heuristic filtering looks, not for specific patterns, but for patterns of behaviors. Positive security model systems model the normal behavior of the web site or web service that they are protecting, and then block connections that do not fit within that model of normal use. URL based and web-form based security checks profile normal use of your web sites, and then control how users interact with your web sites, using both heuristics and positive security to block anomalous or unexpected traffic. Both heuristic and positive security, properly designed and deployed, can catch most attacks that signatures miss. However, they require considerably more resources than do signatures, and you must spend some time configuring them properly to avoid false positives. They are therefore usually used, not as the primary line of defense, but as backups to signatures or other less resource-intensive approaches.

By configuring these advanced protections in addition to signatures, you create a hybrid security model, which enables the App Firewall to provide comprehensive protection against both known and unknown attacks.

How App Firewall works

September 21, 2020

When you install the App Firewall, you create an initial security configuration, which consists of a policy, a profile, and a signatures object. The policy is a rule that identifies the traffic to be filtered, and the profile identifies the patterns and types of behavior to allow or block when the traffic is filtered. The simplest patterns, which are called signatures, are not specified within the profile, but in a signatures object that is associated with the profile.

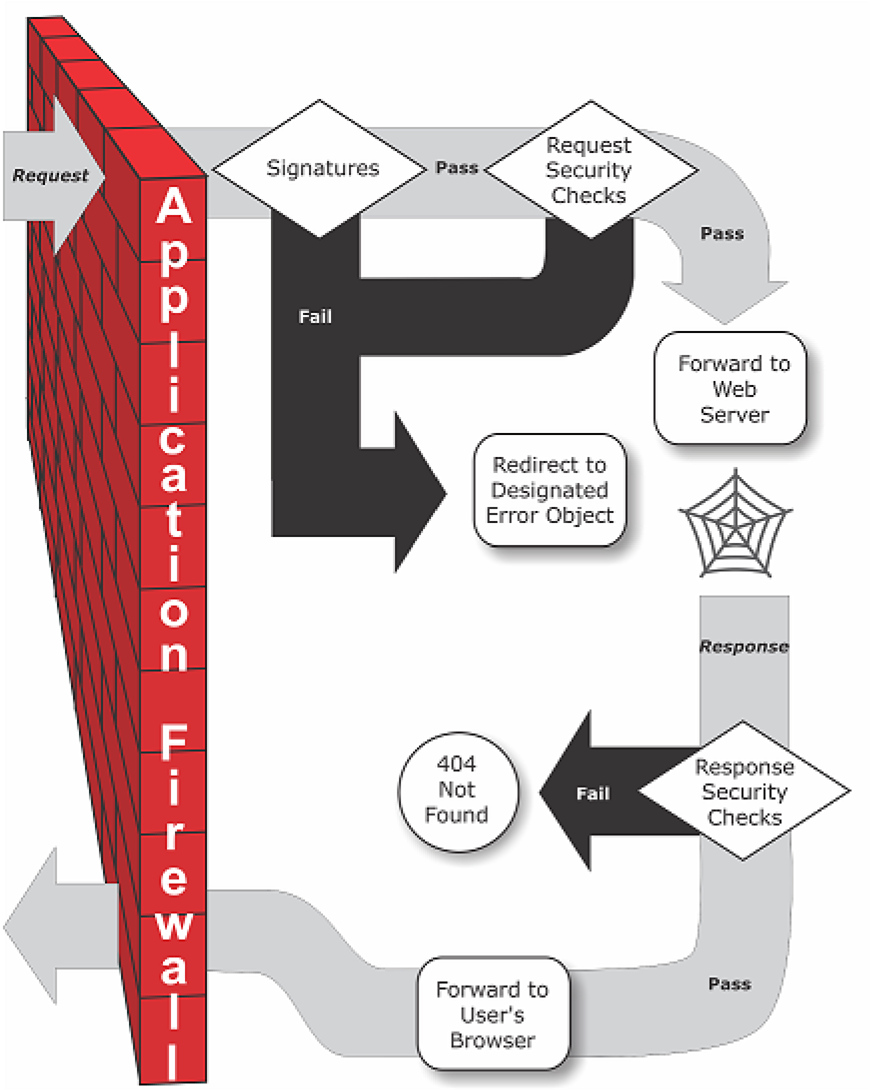
A signature is a string or pattern that matches a known type of attack. The App Firewall contains over a thousand signatures in seven categories, each directed at attacks on specific types of web servers and web content. Citrix updates the list with new signatures as new threats are identified. During configuration, you specify the signature categories that are appropriate for the web servers and content that you need to protect. Signatures provide good basic protection with low processing overhead. If your applications have special vulnerabilities or you detect an attack against them for which no signature exists, you can add your own signatures.

The more advanced protections are called security checks. A security check is a more rigorous, algorithmic inspection of a request for specific patterns or types of behavior that might indicate an attack or constitute a threat to your protected web sites and web services. It can, for example, identify a request that attempts to perform a certain type of operation that might breach security, or a response that includes sensitive private information such as a social security number or credit card number. During configuration, you specify the security checks that are appropriate for the web servers and content that you need to protect. The security checks are restrictive. Many of them can block legitimate requests and responses if you do not add the appropriate exceptions (relaxations) when configuring them. Identifying the needed exceptions is not difficult if you use the adaptive learning feature, which observes normal use of your web site and creates recommended exceptions.

The App Firewall can be installed as either a Layer 3 network device or a Layer 2 network bridge between your servers and your users, usually behind your company's router or firewall. It must be installed in a location where it can intercept traffic between the web servers that you want to protect and the hub or switch through which users access those web servers. You then configure the network to send requests to the App Firewall instead of directly to your web servers, and responses to the App Firewall instead of directly to your users. The App Firewall filters that traffic before forwarding it to its final destination, using both its internal rule set and your additions and modifications. It blocks or renders harmless any activity that it detects as harmful, and then forwards the remaining traffic to the web server. The following figure provides an overview of the filtering process.

> **Note**:
> The figure omits the application of a policy to incoming traffic. It illustrates a security configuration in which the policy is to process all requests. Also, in this configuration, a signatures object has been configured and associated with the profile, and security checks have been configured in the profile.

Figure 1. A Flowchart of App Firewall filtering


Request
Application Firewall
Signatures
Pass
Request Security Checks
Pass
Fail
Forward to Web Server
Redirect to Designated Error Object
Response
404 Not Found
Fail
Response Security Checks
Forward to User's Browser
Pass

As the figure shows, when a user requests a URL on a protected web site, the App Firewall first examines the request to ensure that it does not match a signature. If the request matches a signature, the App Firewall either displays the error object (a web page that is located on the App Firewall appliance and which you can configure by using the imports feature) or forwards the request to the designated error URL (the error page). Signatures do not require as many resources as do security checks, so detecting and stopping attacks that are detected by a signature before running any of the security checks reduces the load on the server.

If a request passes signature inspection, the App Firewall applies the request security checks that have been enabled. The request security checks verify that the request is appropriate for your web site or web service and does not contain material that might pose a threat. For example, security checks examine the request for signs indicating that it might be of an unexpected type, request unexpected content, or contain unexpected and possibly malicious web form data, SQL commands, or scripts. If the request fails a security check, the App Firewall either sanitizes the request and then sends it back to the NetScaler appliance (or NetScaler virtual appliance), or displays the error object. If the request passes the security checks, it is sent back to the NetScaler appliance, which completes any other processing and forwards the request to the protected web server.

When the web site or web service sends a response to the user, the App Firewall applies the response security checks that have been enabled. The response security checks examine the response for leaks of sensitive private information, signs of web site defacement, or other content that should not be present. If the response fails a security check, the App Firewall either removes the content that should not be present or blocks the response. If the response passes the security checks, it is sent back to the NetScaler appliance, which forwards it to the user.

App Firewall features

The basic App Firewall features are policies, profiles, and signatures, which provide a hybrid security model as described in "Known Web Attacks," "Unknown Web Attacks," and "How the App Firewall Works." Of special note is the learning feature, which observes traffic to your protected applications and recommends appropriate configuration settings for certain security checks.

The imports feature manages files that you upload to the App Firewall. These files are then used by the App Firewall in various security checks, or when responding to a connection that matches a security check.

You can use the logs, statistics, and reports features to evaluate the performance of the App Firewall and identify possible needs for additional protections.

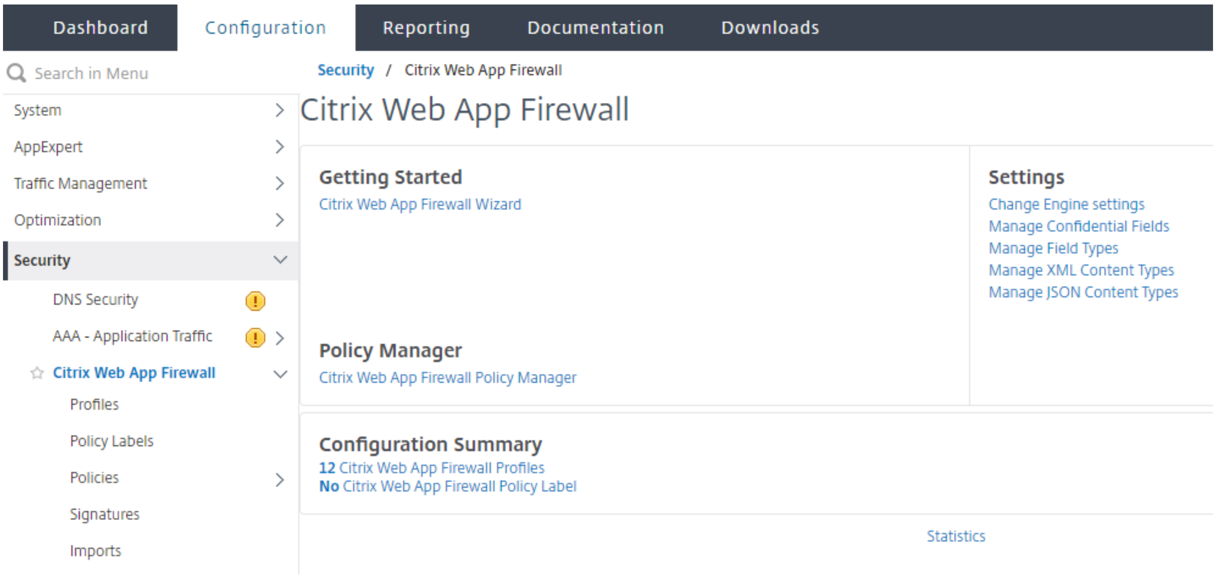
The App Firewall configuration interfaces

September 21, 2020

All hardware and virtual versions of the NetScaler can be configured and managed from the NetScaler command line interface or the web-based GUI. All features of most NetScaler features can be configured using either of these tools. The NetScaler App Firewall is an exception: not all App Firewall configuration tasks can be performed at the command line. Inexperienced users also find the GUI easier to use. In particular, the App Firewall wizard considerably reduces the complexity of configuring the App Firewall. Unlike most NetScaler wizards, the App Firewall wizard can serve as your primary interface to the App Firewall.

The command line interface is a modified UNIX shell based on the FreeBSD bash shell. To configure the App Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. For instructions for using the command line interface, see Command Reference.

The GUI is a web-based GUI interface to the appliance. The Web App Firewall configuration section is found under **Security** > **Web App Firewall**. The below image shows the navigation pane expanded to display the Web App Firewall screens, and in the detail pane the main Web App Firewall screen.


Dashboard
Configuration
Reporting
Documentation
Downloads
Search in Menu
System
AppExpert
Traffic Management
Optimization
Security
DNS Security
AAA - Application Traffic
Citrix Web App Firewall
Profiles
Policy Labels
Policies
Signatures
Imports
Security / Citrix Web App Firewall
Citrix Web App Firewall
Getting Started
Citrix Web App Firewall Wizard
Settings
Change Engine settings
Manage Confidential Fields
Manage Field Types
Manage XML Content Types
Manage JSON Content Types
Policy Manager
Citrix Web App Firewall Policy Manager
Configuration Summary
12 Citrix Web App Firewall Profiles
No Citrix Web App Firewall Policy Label
Statistics

The GUI has two main areas on all screens. The panel on the left, called the navigation pane, contains a navigation tree, with which you navigate to the screens on which you configure the features that are installed on your appliance. The screens to which you navigate appear to the right of the navigation pane, in the details pane.

When you access the GUI, the details pane displays the System Overview screen. If, in the navigation pane, you click plus sign next to the App Firewall folder, the App Firewall node expands to include the main App Firewall elements that you can configure. If you click the first element, Profiles, the details pane displays the configured profiles, if any profiles have been configured. At the bottom of the details pane, you can click Add to configure a new profile. Other buttons at the bottom of the details pane are grayed out until you select an existing profile. Screens for the other elements work in the same way.

If, instead of expanding the App Firewall node, you click the node itself, the details pane displays different options, one of which is the App Firewall wizard, as shown in Figure 1. Citrix recommends that you use the wizard for initial configuration, and many users use it almost exclusively. It includes most of the functionality that is available elsewhere in the GUI.

Configuring the App Firewall

September 21, 2020

You can configure the NetScaler App Firewall by using any of the following methods:

- **App Firewall Wizard.** A dialog box consisting of a series of screens that step you through the configuration process.
- **NetScaler App Interface AppExpert Template.** A NetScaler AppExpert template (a set of configuration settings) that are designed to provide appropriate protection for web sites. This AppExpert template contains appropriate App Firewall configuration settings for protecting many web sites.
- **NetScaler GUI.** The NetScaler web-based configuration interface.
- **NetScaler Command Line Interface.** The NetScaler command line configuration interface.

Citrix recommends that you use the App Firewall Wizard. Most users will find it the easiest method to configure the App Firewall, and it is designed to prevent mistakes. If you have a new NetScaler or VPX that you will use primarily to protect web sites, you may find the Web Interface AppExpert template a better option because it provides a good default configuration, not just for the App Firewall, but for the entire appliance. Both the GUI and the command line interface are intended for experienced users, primarily to modify an existing configuration or use advanced options.

The App Firewall wizard

The App Firewall wizard is a dialog box that consists of several screens that prompt you to configure each part of a simple configuration. The App Firewall then creates the appropriate configuration elements from the information that you give it. This is the simplest and, for most purposes, the best way to configure the App Firewall.

To use the wizard, connect to the GUI with the browser of your choice. When the connection is established, verify that the App Firewall is enabled, and then run the App Firewall wizard, which prompts you for configuration information. You do not have to provide all of the requested information the first time you use the wizard. Instead, you can accept default settings, perform a few relatively straightforward configuration tasks to enable important features, and then allow the App Firewall to collect important information to help you complete the configuration.

For example, when the wizard prompts you to specify a rule for selecting the traffic to be processed, you can accept the default, which selects all traffic. When it presents you with a list of signatures, you can enable the appropriate categories of signatures and turn on the collection of statistics for those signatures. For this initial configuration, you can skip the advanced protections (security checks). The wizard automatically creates the appropriate policy, signatures object, and profile (collectively, the security configuration), and binds the policy to global. The App Firewall then begins filtering connections to your protected websites, logging any connections that match one or more of the signatures that you enabled and collecting statistics about the connections that each signature matches. After the App Firewall processes some traffic, you can run the wizard again and examine the logs and statistics to see if any of the signatures that you have enabled are matching legitimate traffic. After determining which signatures are identifying the traffic that you want to block, you can enable blocking for those signatures. If your website or web service is not complex, does not use SQL, and does not have access to sensitive private information, this basic security configuration will probably provide adequate protection.

You may need additional protection if, for example, your website is dynamic. Content that uses scripts may need protection against cross-site scripting attacks. Web content that uses SQL—such as shopping carts, many blogs, and most content management systems—may need protection against SQL injection attacks. Websites and web services that collect sensitive private information such as social security numbers or credit card numbers may require protection against unintentional exposure of that information. Certain types of web-server or XML-server software may require protection from types of attacks tailored to that software. Another consideration is that specific elements of your websites or web services may require different protection than do other elements. Examining the App Firewall logs and statistics can help you identify the additional protections that you might need.

After deciding which advanced protections are needed for your websites and web services, you can run the wizard again to configure those protections. Certain security checks require that you enter exceptions (relaxations) to prevent the check from blocking legitimate traffic. You can do so manually, but it is usually easier to enable the adaptive learning feature and allow it to recommend the necessary relaxation. You can use the wizard as many times as necessary to enhance your basic security configuration and/or create additional security configurations.

The wizard automates some tasks that you would have to perform manually if you did not use the wizard. It automatically creates a policy, a signatures object, and a profile, and assigns them the name that you provided when you were prompted for the name of your configuration. The wizard also adds your advanced-protection settings to the profile, binds the signatures object to the profile, associates the profile with the policy, and puts the policy into effect by binding it to Global.

A few tasks cannot be performed in the wizard. You cannot use the wizard to bind a policy to a bind point other than Global. If you want the profile to apply to only a specific part of your configuration, you must manually configure the binding. You cannot configure the engine settings or certain other global configuration options in the wizard. While you can configure any of the advanced protection settings in the wizard, if you want to modify a specific setting in a single security check, it may be easier to do so on the manual configuration screens in the GUI.

For more information on using the App Firewall Wizard, see “The App Firewall Wizard.”

NetScaler app interface appExpert template

AppExpert templates are a different and simpler approach to configuring and managing complex enterprise applications. The AppExpert display in the GUI consists of a table. Applications are listed in the left-most column, with the NetScaler features that are applicable to that application appearing each in its own column to the right. (In the AppExpert interface, those features that are associated with an application are called *application units*.) In the AppExpert interface, you configure the interesting traffic for each application, and turn on rules for compression, caching, rewrite, filtering, responder and the App Firewall, instead of having to configure each feature individually.

The Web interface AppExpert template contains rules for the following App Firewall signatures and security checks:

- “**Deny URL check.**” Detects connections to content that is known to pose a security risk, or to any other URLs that you designate.
- “**Buffer Overflow check.**” Detects attempts to cause a buffer overflow on a protected web server.
- “**Cookie Consistency check.**” Detects malicious modifications to cookies set by a protected web site.
- “**Form Field Consistency check.**” Detects modifications to the structure of a web form on a protected web site.
- “**CSRF Form Tagging check.**” Detects cross-site request forgery attacks.
- “**Field Formats check.**” Detects inappropriate information uploaded in web forms on a protected web site.
- “**HTML SQL Injection check.**” Detects attempts to inject unauthorized SQL code.
- “**HTML Cross-Site Scripting check.**” Detects cross-site scripting attacks.

For information on installing and using an AppExpert Template, see “AppExpert Applications and Templates.”

The NetScaler GUI

The GUI is a web-based interface that provides access to all configuration options for the App Firewall feature, including advanced configuration and management options that are not available from any other configuration tool or interface. Specifically, many advanced Signatures options can be configured only in the GUI. You can review recommendations generated by the learning feature only in the GUI. You can bind policies to a bind point other than Global only in the GUI.

For a description of the GUI, see “The App Firewall Configuration Interfaces.” For more information on using the GUI to configure the App Firewall, see “Manual Configuration By Using the GUI.”

For instructions on configuring the App Firewall by using the GUI, see “Manual Configuration By Using the GUI.” For information on the NetScaler GUI, see “The App Firewall Configuration Interfaces.”

The NetScaler command line interface

The NetScaler command line interface is a modified UNIX shell based on the FreeBSD bash shell. To configure the App Firewall from the command line interface, you type commands at the prompt and press the Enter key, just as you do with any other Unix shell. You can configure most parameters and options for the App Firewall by using the NetScaler command line. Exceptions are the signatures feature, many of whose options can be configured only by using the GUI or the App Firewall wizard, and the learning feature, whose recommendations can only be reviewed in the GUI.

For instructions on configuring the App Firewall by using the NetScaler command line, see "Manual Configuration By Using the Command Line Interface."

Enabling App Firewall

September 21, 2020

Before you can create an App Firewall security configuration, you must make sure that the App Firewall feature is enabled.

- If you are configuring a dedicated NetScaler App Firewall appliance or are upgrading an existing NetScaler or VPX, the feature is already enabled. You do not have to perform either of the procedures described here.
- If you have a new NetScaler or VPX, you need to enable the App Firewall feature before you configure it.
- If you are upgrading a NetScaler or VPX from a previous version of the NetScaler operating system to the current version, you might need to enable the App Firewall feature before you configure it.

> **Note**:
>
> If you are upgrading a NetScaler or VPX from a previous version, you might also need to update the licenses on your ADC or VPX before you can enable the App Firewall. Check with your Citrix representative or reseller to obtain the correct license.

You can enable the App Firewall by using the command line or the GUI.

To enable the App Firewall by using the command line interface

At the command prompt, type the following command:

```
enable ns feature AppFW
```

To enable the App Firewall by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Basic Features**.
3. In the **Configure Basic Features** dialog box, check the **App Firewall** check box.
4. Click **OK**.

The App Firewall wizard

September 21, 2020

Unlike most wizards, the NetScaler App App Firewall Wizard is designed not just to simplify the initial configuration process, but also to modify previously created configurations and to maintain your App Firewall setup. A typical user runs the wizard multiple times, skipping some of the screens each time.

The App Firewall Wizard automatically creates profiles, policies, and signatures.

Opening the Wizard

To run the App Firewall Wizard, open the GUI and follow these steps:

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The wizard opens.

For more information about the GUI, see The App Firewall Configuration Interfaces.

The Wizard screens

The App Firewall wizard displays the following screens on a tabular page:

1. Specify Name: on this screen, when creating a new security configuration, specify a meaningful name and the appropriate type (HTML, XML or WEB 2.0) for your profile. The default policy and signatures are auto-generated by using the same name.

Profile Name

The name can begin with a letter, number, or the underscore symbol, and can consist of from 1 to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols. Choose a name that makes it easy for others to tell what content your new security configuration protects.

> **Note:**
>
> Because the wizard uses this name for both the policy and the profile, it is limited to 31 characters. Manually created policies can have names up to 127 characters in length.

When modifying an existing configuration, you select Modify Existing Configuration and then, in the Name drop-down list, select the name of the existing configuration that you want to modify.

> **Note:**
>
> Only policies that are bound to global or to a bind point appear in this list; you cannot modify

> an unbound policy by using the Application Firewall wizard. You must either manually bind it to Global or a bind point, or modify it manually. (For manual modification, in the GUI **Application Firewall > Policies** > **Firewall** pane, select the policy and click **Open**).

Profile Type

You also select a profile type on this screen. The profile type determines the types of advanced protection (security checks) that can be configured. Because certain kinds of content are not vulnerable to certain types of security threats, restricting the list of available checks saves time during configuration. The types of App Firewall profiles are:

- Web Application (HTML). Any HTML-based Web site that does not use XML or Web 2.0 technologies.
- XML Application (XML, SOAP). Any XML-based Web service.
- Web 2.0 Application (HTML, XML, REST). Any Web 2.0 site that combines HTML and XML-based content, such as an ATOM-based site, a blog, an RSS feed, or a wiki.

Note: If you are unsure which type of content is used on your website, you can choose Web 2.0 Application to ensure that you protect all types of web application content.

2. Specify Rule: on this screen, you specify the policy rule (expression) that defines the traffic the current configuration examines. If you create an initial configuration to protect your websites and web services, you can accept the default value, **true**, which selects all web traffic .

If you want this security configuration to examine, not all HTTP traffic that is routed through the appliance, but specific traffic, you can write a policy rule specifying the traffic that you want it to examine. Rules are written in NetScaler expressions language, which is a fully functional object-oriented programming language. For more information, see Configure a Custom Policy Expression.

Note: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current users who want to migrate their existing configurations to the NetScaler cluster must migrate any policies that contain classic expressions to the default expressions syntax.

- For a simple description of using the NetScaler expressions syntax to create App Firewall rules, and a list of useful rules, see “Firewall Policies.”
- For a detailed explanation of how to create policy rules in NetScaler expressions syntax, see “Policies and Expressions.”

4. Select Signatures: on this screen, you select the categories of signatures that you want to use to protect your web sites and web services.

This is not a mandatory step, and you can skip it if you want to and go to the **Specify Deep Protections** screen. If the Select Signatures screen is skipped, only a profile and associated policies are created, and the signatures are not created.

You can select **Create New Signature** or **Select Existing Signature**.

If you are creating a new security configuration, the signature categories that you select are enabled, and by default they are recorded in a new signatures object. The new signatures object is assigned the same name that you entered on the Specify name screen as the name of the security configuration.

If you have previously configured signatures objects and want to use one of them as the signatures object associated with the security configuration that you are creating, click **Select Existing Signature** and select a signatures object from the Signatures list.

If you are modifying an existing security configuration, you can click Select Existing Signature and assign a different signatures object to the security configuration.

If you click Create New Signature, you can choose the edit mode as **Simple** or **Advanced**.

1. **Specify Signature Protections (Simple mode)**

The simple mode allows for easy configuration of the signature, with a preset list of protection definitions for common applications such as IIS (Internet Information Server), PHP and ActiveX. The default categories in Simple mode are:

- CGI. Protection against attacks on web sites that use CGI scripts in any language, including PERL scripts, Unix shell scripts, and Python scripts.
- Cold Fusion. Protection against attacks on web sites that use the Adobe Systems® ColdFusion® Web development platform.
- FrontPage. Protection against attacks on web sites that use the Microsoft® FrontPage® Web development platform.
- PHP. Protection against attacks on web sites that use the PHP open-source Web development scripting language.
- Client side. Protection against attacks on client-side tools used to access your protected web sites, such as Microsoft Internet Explorer, Mozilla Firefox, the Opera browser, and the Adobe Acrobat Reader.
- Microsoft IIS. Protection against attacks on Web sites that run the Microsoft Internet Information Server (IIS)
- Miscellaneous. Protection against attacks on other server-side tools, such as Web servers and database servers.

On this screen, you select the actions associated with the signature categories that you selected on the Select Signatures screen. The actions that you can configure are:

- Block
- Log

- Stats

By default the Log and Stats actions are enabled but not the Block action. To configure actions, click **Settings**. You can change the action settings of all the selected categories by using the **Action** drop-down menu.

1. **Specify Signature Protections (Advanced mode)**

The advanced mode allows for more granular control over the signature definitions and provides significantly more information. Use the advanced mode if you want complete control over signature definition.

The contents of this screen are the same as the contents of the Modify Signatures Object dialog box, as described in "Configuring or Modifying a Signatures Object." In this screen, you can configure actions either by clicking the **Actions** drop-down menu or the actions menu, which appears as a cirle with three dots.

7. Specify Deep Protections: on this screen, you choose the advanced protections (also called security checks or simply checks) that you want to use to protect your web sites and web services. Which checks are available depends on the profile type that you chose on the Specify Name screen. All checks are available for Web 2.0 Application profiles.

For more information, see Overview of Security Checks and see "Advanced Form Protections Checks."

You configure the actions for the advanced protections that you have enabled.The actions that you can configure are:

- Block: blocks connections that match the signature. Disabled by default.
- Log: logs connections that match the signature for later analysis. Enabled by default.
- Stats: maintains statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.
- Learn. Observe traffic to this website or web service, and use connections that repeatedly violate this check to generate recommended exceptions to the check, or new rules for the check. Available only for some checks. For more information about the learning feature see "Configuring and Using the Learning Feature," and how learning works and how to configure exceptions (relaxations) or deploy learned rules for a check, see "Manual Configuration By Using the GUI."

To configure actions, select the protection by clicking the check box, and then click **Action Settings** to select the required actions. Select other parameters, if required, and then click **OK** to close the Action Settings window.

To view all logs for a specific check, select that check, and then click **Logs** to display the Syslog Viewer, as described in "App Firewall Logs." If a security check is blocking legitimate access to your protected web site or web service, you can create and implement a relaxation for that security check by selecting a log that shows the unwanted blocking, and then clicking **Deploy**.

After you completing specifying Action Settings, click **Finish** to complete the wizard.

Following are four procedures that show how to perform specific types of configuration by using the App Firewall wizard.

Create a New Configuration

Follow these steps to create a new firewall configuration and signature objects, by using the Applicaiton Firewall wizard.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.**The wizard opens.
3. On the **Specify Name** screen, select **Create New Configuration.**
4. In the **Name** field, type a name, and then click **Next**.
5. In the **Specify Rule** screen, click **Next** again.
6. In the **Select Signatures** screen, select **Create New Signature** and **Simple** as the edit mode, and then click **Next**.
7. In the **Specify Signature Protections** screen, configure the required settings. For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see "Signatures."
8. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
9. When you complete, click **Finish** to close the Application Firewall wizard.

Modify an Existing Configuration

Follow these steps to modify an existing configuration and existing signature categories.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**. The wizard opens.
3. On the **Specify Name** screen, select Modify Existing Configuration and, in the **Name** drop-down list, choose the security configuration that you created during new configuration, and then click **Next**.
4. In the **Specify Rule** screen, click Next to keep the default value "true." If you want to modify the rule, follow the steps described in "Configure a Custom Policy Expresssion."
5. In the **Select Signatures** screen, click **Select Existing Signature**. From the **Existing Signature** drop-down menu, select the appropriate option, and then click **Next**. The advanced signature

protection screen appears.

Note: If you select an existing signature, the default edit mode for signature protected is advanced.

6. In the Specify Signature Protections screen, configure the required settings and click **Next**. For more information about which signatures to consider for blocking and how to determine when you can safely enable blocking for a signature, see "Signatures."
7. In the **Specify Deep Protections** screen, configure the settings and click **Next**.
8. After you complete, click **Finish** to close the **App Firewall Wizard**.

Create a New Configuration without Signatures

Follow these steps to use the Application Firewall Wizard to skip the Select Signatures screen and create a new configuration with just the profile and the associated policies but without any signatures.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.The wizard opens.
3. On the **Specify Name** screen, select **Create New Configuration**.
4. In the **Name** field, type a name, and then click **Next**.
5. In the **Specify Rule** screen, click **Nex**t again.
6. In the **Select Signatures** screen, click **Skip.**
7. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
8. When you complete, click **Finish** to close the Application Firewall Wizard.

Configure a Custom Policy Expression

Follow these steps to use the Application Firewall Wizard to create a specialized security configuration to protect only specific content. In this case, you create a new security configuration instead of modifying the initial configuration. This type of security configuration requires a custom rule, so that the policy applies the configuration to only the selected Web traffic.

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Getting Started**, click **Application Firewall Wizard**.
3. On the Specify Name screen, type a name for your new security configuration in the Name text box, select the type of security configuration from the Type drop-down list, and then click **Next**.
4. On the **Specify Rule** screen, enter a rule that matches only that content that you want this web application to protect. Use the **Frequently Used Expressions** drop-down list and the **Expression Editor** to create a custom expression. When you complete, click **Next**.
5. In the **Select Signatures** screen, select the edit mode, and then click **Next**.
6. In the **Specify Signature Protections** screen, configure the required settings.

7. In the **Specify Deep Protections** screen configure the required actions and parameters in **Action Settings**.
8. When you complete, click **Finish** to close the **Application Firewall Wizard**.

Manual configuration

September 21, 2020

If you want to bind a profile to a bind point other than Global, you must manually configure the binding. Also, certain security checks require that you either manually enter the necessary exceptions or enable the learning feature to generate the exceptions that your Web sites and Web services need. Some of these tasks cannot be performed by using the App Firewall wizard.

If you are familiar with how the App Firewall works and prefer manual configuration, you can manually configure a signatures object and a profile, associate the signatures object with the profile, create a policy with a rule that matches the web traffic that you want to configure, and associate the policy with the profile. You then bind the policy to Global, or to a bind point, to put it into effect, and you have created a complete security configuration.

For manual configuration, you can use the GUI (a graphical interface) or the command line. Citrix recommends that you use the GUI. Not all configuration tasks can be performed at the command line. Certain tasks, such as enabling signatures and reviewing learned data, must be done in the GUI. Most other tasks are easier to perform in the GUI.

Replicating configuration

When you use the GUI (GUI) or the command line interface (CLI) to manually configure the App Firewall, the configuration is saved in the /nsconfig/ns.conf file. You can use the commands in that file to replicate the configuration on another appliance. You can cut and paste the commands into the CLI one by one, or you can save multiple commands in a text file in the /var/tmp folder and run them as a batch file. Following is an example of running a batch file containing commands copied from the /nsconfig/ns.conf file of a different appliance:

```
> batch -f /var/tmp/appfw_add.txt
```

> **Warning**
>
> Import commands are not saved in the ns.conf file. Before running commands from the ns.conf file to replicate the configuration on another appliance, you must import all the objects used in the configuration (for example, signatures, error page, WSDL, and Schema) to the appliance on which you will replicate the configuration. The add command to add an App Firewall profile

> saved in an ns.conf file might include the name of an imported object, but such a command might fail when executed on another appliance if the referenced object does not exist on that appliance.

Manual configuration by using the GUI

July 19, 2021

If you need to configure the App Firewall feature manually, Citrix recommends that you use the GUI.

To create and configure a signatures object

Before you can configure the signatures, you must create a signatures object from the appropriate default signatures object template. Assign the copy a new name, and then configure the copy. You cannot configure or modify the default signatures objects directly. The following procedure provides basic instructions for configuring a signatures object. For more detailed instructions, see "Manually Configuring the Signatures Feature."

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to use as a template, and then click **Add**.

   Your choices are:

   - *Default Signatures. Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - *XPath Injection. Contains all of the items in the Default Signatures, and in addition, contains the XPath injection rules.
3. In the **Add Signatures Object** dialog box, type a name for your new signatures object, click OK, and then click Close. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
4. Select the signatures object that you created, and then click **Open**.
5. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

   As you modify these options, the results that you specify are displayed in the Filtered Results window at the right. For more information about the categories of signatures, see "Signatures."
6. In the **Filtered Results** area, configure the settings for a signature by selecting and clearing the appropriate checkboxes.

7. When finished, finished, click **Close**.

To create an App Firewall profile by using the GUI

Creating an App Firewall profile requires that you specify only a few configuration details.

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, click **Add**.
3. In the C**reate App Firewall** Profile dialog box, type a name for your profile.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Choose the profile type from the drop-down list.
5. Click **Create**, and then click **Close**.

To configure an App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**.
3. In the **Configure App Firewall Profile** dialog box, on the **Security Checks** tab, configure the security checks.

   - To enable or disable an action for a check, in the list, select or clear the checkbox for that action.
   - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.

     You can also select a check and, at the bottom of the dialog box, click Open to display the Configure Relaxation dialog box or Configure Rule dialog box for that check. These dialog boxes also vary from check to check. Most of them include a Checks tab and a General tab. If the check supports relaxations or user-defined rules, the Checks tab includes an Add button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click Open to modify it.

   - To review learned exceptions or rules for a check, select the check, and then click Learned Violations. In the Manage Learned Rules dialog box, select each learned exception or rule in turn.

    - To edit the exception or rule, and then add it to the list, click **Edit & Deploy**.
    - To accept the exception or rule without modification, click **Deploy**.
    - To remove the exception or rule from the list, click **Skip**.

  - To refresh the list of exceptions or rules to be reviewed, click **Refresh**.
  - To open the **Learning Visualizer** and use it to review learned rules, click **Visualizer**.
  - To review the log entries for connections that matched a check, select the check, and then click **Logs**. You can use this information to determine which checks are matching attacks so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "Logs, Statistics, and Reports."
  - To completely disable a check, in the list, clear all of the checkboxes to the right of that check.

4. On the **Settings** tab, configure the profile settings.

    - To associate the profile with the set of signatures that you previously created and configured, under **Common Settings**, choose that set of signatures in the Signatures drop-down list.

      > **Note**:
      >
      > You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.

    - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.

      > **Note**:
      >
      > You must first upload the error object that you want to use in the Import pane.

    - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types.

5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile. For more information, see Configure and Learning feature..
6. Click **OK** to save your changes and return to the Profiles pane.

Configuring an App Firewall rule or relaxation

You configure two different types of information in this dialog box, depending upon which security check you are configuring. In the majority of cases, you configure an exception (or relaxation) to the security check. If you are configuring the Deny URL check or the Field Formats check, you configure an addition (or rule). The process for either of these is the same.

To configure a relaxation or rule by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the **Profiles** pane, select the profile you want to configure, and then click **Edit**.
3. In the **Configure App Firewall Profile** dialog box, click the **Security Checks** tab. The **Security Checks** tab contains the complete list of App Firewall security checks, also called *advanced protections* in some places.
4. In the **Security Checks** tab, click the check that you want to configure, and then click Open. The **Modify Check** dialog box for the check that you chose is displayed, with the Checks tab selected. The Checks tab contains a list of existing relaxations or rules for this check. The list might be empty if you have not either manually added any relaxations or approved any relaxations that were recommended by the learning engine. Beneath the list is a row of buttons that allow you to add, modify, delete, enable, or disable the relaxations on the list.
5. To add or modify a relaxation or a rule, do one of the following:
   - To add a new relaxation, click **Add**.
   - To modify an existing relaxation, select the relaxation that you want to modify, and then click **Open**.

   The **Add Check Relaxation** or **Modify Check Relaxation** dialog box for the selected check is displayed. Except for the title, these dialog boxes are identical.
6. Fill in the dialog box as described below. The dialog boxes for each check are different; the list below covers all elements that might appear in any dialog box.
   - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
   - **Attachment Content Type**—The Content-Type attribute of an XML attachment. In the text area, enter a regular expression that matches the Content-Type attribute of the XML attachments to allow.
   - **Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
   - **Cookie**—In the text area, enter a PCRE-format regular expression that defines the cookie.
   - **Field Name**—A web form field name element may be labeled Field Name, Form Field, or another similar name. In the text area, enter a PCRE-format regular expression that defines the name of the form field.

- **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the web form.
- **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the web form is delivered.
- **Name**—An XML element or attribute name. In the text area, enter a PCRE-format regular expression that defines the name of the element or attribute.
- **URL**—A URL element may be labeled Action URL, Deny URL, Form Action URL, Form Origin URL, Start URL, or simply URL. In the text area, enter a PCRE-format regular expression that defines the URL.
- **Format**—The format section contains multiple settings that include list boxes and text boxes. Any of the following can appear:
  - **Type**—Select a field type in the Type drop-down list. To add a new field type definition, click Manage—
  - **Minimum Length**—Type a positive integer that represents the minimum length in characters if you want to force users to fill in this field. Default: 0 (Allows field to be left blank.)
  - **Maximum length**—To limit the length of data in this field, type a positive integer that represents the maximum length in characters. Default: 65535
- **Location**—Choose the element of the request that your relaxation will apply to from the drop-down list. For HTML security checks, the choices are:
  - FORMFIELD—Form fields in web forms.
  - HEADER—Request headers.
  - COOKIE—Set-Cookie headers.

  For XML security checks, the choices are:

  - ELEMENT—XML element.
  - ATTRIBUE—XML attribute.
- **Maximum Attachment Size**—The maximum size in bytes allowed for an XML attachment.
- **Comments**—In the text area, type a comment. Optional.

Note: For any element that requires a regular expression, you can type the regular expression, use the
Regex Tokens menu to insert regular expression elements and symbols directly into the text box, or click **Regex Editor** to open the **Add Regular Expression** dialog box, and use it to construct the expression.

7. To remove a relaxation or rule, select it, and then click **Remove**.
8. To enable a relaxation or rule, select it, and then click **Enable**.

9. To disable a relaxation or rule, select it, and then click **Disable**.
10. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click **Visualizer**, and use the display tools.

    > **Note**:
    >
    > The **Visualizer** button does not appear on all check relaxation dialog boxes.

11. To review learned rules for this check, click Learning and perform the steps in To configure and use the Learning feature
12. Click **OK**.

To configure the Learning feature by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the **Profiles** pane, select the profile, and then click **Edit**.
3. Click the **Learning** tab. At the top of the **Learning** tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:
    - **Minimum number threshold.** Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1
    - **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0
5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click Remove All Learned Data.

    > **Note**:
    >
    > This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

6. To restrict the learning engine to traffic from a specific set of IPs, click **Trusted Learning Clients**, and add the IP addresses that you want to use to the list.

   a) To add an IP address or IP address range to the Trusted Learning Clients list, click **Add**.
   b) In the **Add Trusted Learning Clients** dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
   c) In the Comments text area, type a comment that describes this IP address or range.
   d) Click Create to add your new IP address or range to the list.
   e) To modify an existing IP address or range, click the IP address or range, and then click Open. Except for the name, the dialog box that appears is identical to the **Add Trusted Learning Clients** dialog box.
   f) To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click **Disable** or **Enable**, as appropriate.
   g) To remove an IP address or range completely, click the IP address or range, and then click **Remove**.

7. Click **Close** to return to the **Configure application firewall Profile** dialog box.

8. Click **Close** to close the **Configure application firewall Profile** dialog box, and return to the **application firewall Profile** screen.

To create and configure a policy by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Policies**.

2. In the details pane, do one of the following:

   - To create a new firewall policy, click **Add**. The **Create App Firewall Policy** is displayed.
   - To edit an existing firewall policy, select the policy, and then click **Edit**.

   The **Create App Firewall Policy** or **Configure App Firewall Policy** is displayed.

3. If you are creating a new firewall policy, in the **Create App Firewall Policy** dialog box, Policy Name text box, type a name for your new policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

   If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the **Profile** drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking **Modify**.

5. In the Expression text area, create a rule for your policy.

   - You can type a rule directly into the text area.
   - You can click Prefix to select the first term for your rule, and follow the prompts.
   - You can click Add to open the Add Expression dialog box, and use it to construct the rule.

6. Click **Create** or **OK**, and then click **Close**.

To create or configure an App Firewall rule (expression)

The policy rule, also called the *expression*, defines the web traffic that the App Firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), App Firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the App Firewall wizard or the NetScaler GUI to create your policy rule:
   - If you are configuring a policy in the App Firewall wizard, in the navigation pane, click **application firewall**, then in the details pane click **application firewall** Wizard, and then navigate to the **Specify Rule** screen.
   - If you are configuring a policy manually, in the navigation pane, expand application firewall, **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, the Create application firewall Profile dialog box, or the Configure Application Firewall Profile dialog box, click **Prefix**, and then choose the prefix for your expression from the drop-down list. Your choices are:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the App Firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.
3. Choose your next term.

   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The App Firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you have decided which term you want, double-click it to insert it into the Expression window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(“”), type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

   Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     ```
     HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     <!--NeedCopy-->
     ```

     For shopping.example.com, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     ```
     HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     <!--NeedCopy-->
     ```

     For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

   - **Specific type of content: GIF images.** To match GIF format images:

     ```
     HTTP.REQ.URL.ENDSWITH(".png")
     <!--NeedCopy-->
     ```

     To match other format images, substitute another string in place of .png.

   - **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

     ```
     HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
     <!--NeedCopy-->
     ```

     To match all JavaScripts with .js extensions:

     ```
     HTTP.REQ.URL.ENDSWITH(".js")
     <!--NeedCopy-->
     ```

For more information about creating policy expressions, see "Policies and Expressions."

> **Note**:
> If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the GUI:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

If entered at the command line, however, you must type this instead:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

To add a firewall rule (expression) by using the Add Expression dialog box

The **Add Expression** dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the App Firewall wizard or the NetScaler GUI:
    - If you are configuring a policy in the **App Firewall** wizard, in the navigation pane, click **App Firewall**, then in the details pane click **App Firewall** Wizard, and then navigate to the **Specify Rule** screen.
    - If you are configuring a policy manually, in the navigation pane, expand **App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, in the **Create App Firewall Profile** dialog box, or in the **Configure App Firewall Profile** dialog box, click **Add**.
3. In the **Add Expression** dialog box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
    - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
    - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
    - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
    - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the App Firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the Preview Expression window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The Preview Expression window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click OK to close the Add Expression dialog box. Your expression is inserted into the Expression text area.

To bind an App Firewall policy by using the GUI

1. Do one of the following:
    - Navigate to **Security** > **App Firewall**, and in the details pane, click **application firewall policy manager**.
    - Navigate to **Security > Citrix Web App Firewall > Policies > Firewall Policies**, and in the details pane, click **Policy Manager**.
2. In the **Application Firewall Manager** dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
    - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
    - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
    - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
    - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.
    - **Policy Label.** Policies that are bound to a policy label process traffic that the policy label

routes to them. The policy label controls the order in which policies are applied to this traffic.
- **None.** Do not bind the policy to any bind point.

3. Click **Continue**. A list of existing App Firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
    - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select **Regenerate Priorities** to renumber the priorities evenly.
    - To modify the policy expression, double click that field to open the **Configure App Firewall Policy** dialog box, where you can edit the policy expression.
    - To set the Goto Expression, double click field in the **Goto Expression** column heading to display the drop-down list, where you can choose an expression.
    - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression.
6. Repeat steps 3 through 6 to add any additional App Firewall policies you want to globally bind.
7. Click **OK**. A message appears in the status bar, stating that the policy has been successfully bound.

Manual configuration By using the command line interface

September 21, 2020

You can configure Web App Firewall features from the **NetScaler** command interface. However, there are exceptions such as enabling signatures from the command interface. There are over 1,000 default signatures in seven categories. The task is too complex for the command line interface. You can configure the check actions and parameters for security checks from the command interface, but cannot manually add relaxations. When you configure Adaptive Learning feature and enable it from the command interface, you cannot review learned relaxations or learned rules but approve or skip them. The command interface is intended for advanced users who are familiar with NetScaler appliance and the Web App Firewall feature.

To manually configure Web App Firewall by using the **NetScaler** command interface, use a telnet or secure shell client.

To create a profile by using the command line interface

At the command prompt, type the following commands:

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`

- `save ns config`

Example

The following example is a profile named pr-basic, with basic defaults with a profile type HTML. This profile type is created to protect an HTML website.

```
add appfw profile pr-basic -defaults basic
set appfw profile pr-basic -type HTML
save ns config
<!--NeedCopy-->
```

To configure a profile by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> <arg1> [<arg2> ...]` where `<arg1>` represents a parameter and `<arg2>` represents either another parameter or the value to assign to the parameter represented by `<arg1>`. For descriptions of the parameters to use when configuring specific security checks, see Advanced Protections and its subtopics. For descriptions of the other parameters, see "Parameters for Creating a Profile."
- `save ns config`

Example

The following example shows how to configure an HTML profile created with basic defaults to protect a simple HTML-based website. The profile turns on logging and maintenance of statistics for most security checks, but enables blocking only for those checks that have low false positive rates and requires no special configuration. It turns on transformation of unsafe HTML and unsafe SQL to prevent attacks. The profile also blocks requests to your websites. With logging and statistics enabled, you can later review the logs to determine whether to enable blocking for a specific security check.

```
set appfw profile -startURLAction log stats
set appfw profile -denyURLAction block log stats
set appfw profile -cookieConsistencyAction log stats
set appfw profile -crossSiteScriptingAction log stats
set appfw profile -crossSiteScriptingTransformUnsafeHTML ON
set appfw profile -fieldConsistencyAction log stats
set appfw profile -SQLInjectionAction log stats
set appfw profile -SQLInjectionTransformSpecialChars ON
set appfw profile -SQLInjectionOnlyCheckFieldsWithSQLChars ON
set appfw profile -SQLInjectionParseComments checkall
set appfw profile -fieldFormatAction log stats
set appfw profile -bufferOverflowAction block log stats
set appfw profile -CSRFtagAction log stats
save ns config
<!--NeedCopy-->
```

To create and configure a policy

At the command prompt, type the following commands:

- `add appfw policy <name> <rule> <profile>`
- `save ns config`

Example

The following example is a policy with a rule that intercepts all traffic from a client, `blog.example.com`, and associates a Web App Firewall policy to the profile. The policy protects the blog hosted on a specific host name.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
    ")" pr-blog
<!--NeedCopy-->
```

To bind a Web App Firewall policy

At the command prompt, type the following commands:

- `bind appfw global <policyName> <priority>`
- `save ns config`

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10
save ns config
<!--NeedCopy-->
```

Signatures

September 21, 2020

The App Firewall signatures function provides specific, configurable rules to simplify the task of protecting your web sites against known attacks. A signature represents a pattern that is a component of a known attack on an operating system, web server, website, XML-based web service, or other resource. A rich set of preconfigured App Firewall built-in or Native rules offers an easy to use security solution, leveraging the power of pattern matching to detect attacks and protect against application vulnerabilities.

You can create your own signatures or use the signatures in the built-in templates. The App Firewall has two built-in templates:

- **Default Signatures**: This template contains a preconfigured list of over 1,300 signatures, in addition to a complete list of SQL injection keywords, SQL special strings, SQL transform rules, and SQL wildchar characters. It also contains denied patterns for cross-site scripting, and allowed attributes and tags for cross-site scripting. This is a read-only template. You can view the contents, but you cannot add, edit, or delete anything in this template. To use it, you must make a copy. In your own copy, you can enable the signature rules that you want to apply to your traffic, and specify the actions to be taken when the signature rules match the traffic.

The App Firewall signatures are derived from the rules published by Snort, which is an open source intrusion prevention system capable of performing real-time traffic analysis to detect a variety of attacks and probes.

- ***Xpath Injection Patterns**: This template contains a preconfigured set of literal and PCRE keywords and special strings that are used to detect XPath (XML Path Language) injection attacks.

Blank Signatures: In addition to making a copy of the built-in *Default Signatures template, you can use a blank signatures template to create a new signature object. The signature object that you create with the blank signatures option does not have any native signature rules, but, just like the *Default template, it has all the SQL/XSS built-in entities.

External-Format Signatures: The App Firewall also supports the use of external format signatures. You can import the scan files of the third party scan tools by using the XSLT files that are supported by the NetScaler App Firewall. A set of built-in XSLT files are available for the following scan tools to translate these external format files to the Native format:

- Cenzic
- Deep Security for Web Apps
- IBM AppScan Enterprise
- IBM AppScan Standard.
- Qualys
- Whitehat
- Hewlett Packard Enterprise WebInspect

Protection options for your application

Tighter security increases processing overhead. Signatures provide the following deployment options to help you to optimize the protection of your applications:

- **Negative Security Model**: With the negative security model, you use a rich set of preconfigured signature rules to leverage the power of pattern matching to detect attacks and protect against application vulnerabilities. You block only what you don't want and allow the rest. You can add your own signature rules, based on the specific security needs of your applications, to design your own customized security solutions.
- **Hybrid security Model**: In addition to using signatures, you can use positive security checks to create a configuration ideally suited for your applications. Use signatures to block what you don't want, and use positive security checks to enforce what is allowed.

To protect your application by using signatures, you must configure one or more profiles to use your signatures object. In a hybrid security configuration, the SQL injection and Cross-Site scripting patterns, and the SQL transformation rules, in your signatures object are used not only by the signature rules, but also by the positive security checks configured in the App Firewall profile that is using the signatures object.

The App Firewall examines the traffic to your protected web sites and web services to detect traffic that matches a signature. A match is triggered only when every pattern in the rule matches the traffic. When a match occurs, the specified actions for the rule are invoked. You can display an error page or error object when a request is blocked. Log messages can help you to identify attacks being launched against your application. If you enable statistics, the App Firewall maintains data about requests that match an App Firewall signature or security check.

If the traffic matches both a signature and a positive security check, the more restrictive of the two actions is enforced. For example, if a request matches a signature rule for which the block action is disabled, but the request also matches an SQL Injection positive security check for which the action is block, the request is blocked. In this case, the signature violation might be logged as <`not blocked`>, although the request is blocked by the SQL injection check.

Customization: If necessary, you can add your own rules to a signatures object. You can also customize the SQL/XSS patterns. The option to add your own signature rules, based on the specific security needs of your applications, gives you the flexibility to design your own customized security solutions. You block only what you don't want and allow the rest. A specific fast-match pattern in a specified location can significantly reduce processing overhead to optimize performance. You can add, modify, or remove SQL injection and cross-site scripting patterns. Built-in RegEx and expression editors help you configure your patterns and verify their accuracy.

Auto-update: You can manually update the signature object to get the latest signature rules, or you can leverage the auto-update feature so that the App Firewall can automatically update the signatures from the cloud-based App Firewall updates service.

> **Note**
>
> If new signature rules are added during auto-update, they are disabled by default. You should periodically review the updated signatures and enable the newly added rules that are pertinent for protecting your applications.
>
> You must configure CORS to host signatures on IIS servers.

Getting started

Using Citrix signatures to protect your application is quite easy and can be accomplished in a few simple steps:

1. Add a signature object.
    - You can use the Wizard that prompts you to create the entire App Firewall configuration, including adding the profile and policy, selecting and enabling signatures, and specifying actions for signatures and positive security checks. The signatures object is created automatically.
    - You can create a copy of the signatures object from the *Default Signatures template, use a blank template to create a new signature with your own customized rules, or add an external format signature. Enable the rules and configure the actions that you want to apply.
2. Configure the target App Firewall profile to use this signatures object.
3. Send traffic to validate the functionality

Highlights

- The *Default signatures object is a template. It cannot be edited or deleted. To use it, you must create a copy. In your own copy, you can enable the rules and the desired action for each rule as required for your application. To protect the application, you must configure the target profile to use this signature.
- Processing signature patterns has overhead. Try to enable only those signatures that are applicable for protecting your application, rather than enabling all signature rules.
- Every pattern in the rule must match to trigger a signature match.
- You can add your own customized rules to inspect incoming requests to detect various types of attacks, such as SQL injection or cross-site scripting attacks. You can also add rules to inspect the responses to detect and block leakage of sensitive information such as credit card numbers.
- You can make a copy of an existing signatures object and tweak it, by adding or editing rules and SQL/XSS patterns, to protect another application.
- You can use auto-update to download the latest version of the App Firewall default rules without need for ongoing monitoring to check for the availability of the new update.

- A signature object can be used by more than one profile. Even after you have configured one or more profile(s) to use a signature object, you can still enable or disable signatures or change the action settings. You can manually create and modify your own custom signature rules. The changes will apply to all the profiles that are currently configured to use this signature object.
- You can configure signatures to detect violations in various types of payloads, such as HTML, XML, JSON, and GWT.
- You can export a configured signatures object and import it to another NetScaler appliance for easy replication of your customized signature rules.

Signatures are patterns that are associated with a known vulnerability. You can use signature protection to identify the traffic that attempts to exploit these vulnerabilities, and take specific actions.

Signatures are organized into categories. You can optimize the performance and reduce the processing overhead by enabling only the rules in the categories that are appropriate for protecting your application.

Manually configuring the signatures feature

September 21, 2020

To use signatures to protect your web sites, you must review the rules, and enable and configure the ones that you want to apply. The rules are disabled by default. Citrix recommends that you enable all rules that are applicable to the type of content that your web site uses.

To manually configure the signatures feature, use a browser to connect to the GUI. Then, create a signatures object from a built-in template, an existing signatures object, or by importing a file. Next, configure the new signatures object as explained in Configuring or Modifying a Signatures Object.

Adding or removing a signatures object

September 22, 2020

You can add a new signatures object to the App Firewall by:

- Copying a built-in template.
- Copying an existing signatures object.
- Importing a signatures object from an external file.

You must use the GUI to copy a template or existing signatures object. You can use either the GUI or the command line to import a signatures object. You can also use either the GUI or the command line to remove a signatures object.

To create a signatures object from a template

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to use as a template.

   Your choices are:

   - **Default Signatures.** Contains the signatures rules, the SQL injection rules, and the cross-site scripting rules.
   - **XPath Injection.** Contains the XPath injection patterns.
   - **Any existing signatures object.**

   > **Attention:**
   >
   > If you do not choose a signatures type to use as a template, the App Firewall will prompt you to create signatures from scratch.

3. Click **Add**.
4. In the Add Signatures Object dialog box, type a name for your new signatures object, and then click OK. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols.
5. Click **Close**.

To create a signatures object by importing a file

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, click **Add**.
3. In the **Add Signatures Object** dialog box, select the format of the signatures you want to import.
   - To import a NetScaler format signatures file, select the **Native Format** tab.
   - To import an external signatures format file, select the **External Format** tab.
4. Choose the file that you want to use to create your new signatures object.
   - To import a native NetScaler format signatures file, in the Import section select either Import from Local File or Import from URL, then type or browse to the path or URL to the file.
   - To import a Cenzic, IBM AppScan, Qualys, or Whitehat format file, in the XSLT section select Use Built-in XSLT File, Use Local File, or Reference from URL. Next, if you chose Use Built-in XSLT File, select the appropriate file format from the drop-down list. If you chose Use Local File or Reference from URL, then type or browse to the path or URL to the file.
5. Click **Add**, and then click **Close**.

To create a signatures object by importing a file by using the command line

At the command prompt, type the following commands:

- `import appfw signatures <src> <name> [-xslt <string>] [-comment <string>] [-overwrite] [-merge] [-sha1 <string>]`
- `save ns config`

Example #1

The following example creates a new signatures object from a file named signatures.xml and assigns it the name MySignatures.

```
import appfw signatures signatures.xml MySignatures
save ns config
<!--NeedCopy-->
```

To remove a signatures object by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to remove.
3. Click **Remove**.

To remove a signatures object by using the command line

At the command prompt, type the following commands:

- `rm appfw signatures <name>`
- `save ns config`

Configuring or modifying a signatures object

September 22, 2020

You configure a signatures object after creating it, or modify an existing signatures object, to enable or disable signature categories or specific signatures, and configure how the App Firewall responds when a signature matches a connection.

To configure or modify a signatures object

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.

2. In the details pane, select the signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, set the **Display Filter Criteria** options at the left to display the filter items that you want to configure.

As you modify these options, the results that you requested are displayed in the Filtered Results window at the right.

```
-  To display only selected categories of signatures, check or clear the appropriate signature-category check boxes. The signature categories are:

    | Name        | Type of Attack that this Signature Protects Against         |
    | ----------- | ----------------------------------------------------------------------- |
    | cgi         | CGI scripts. Includes Perl and UNIX shell scripts.          |
    | client      | Browsers and other clients.                                  |
    | coldfusion  | Web sites that use the Adobe Systems ColdFusion application server.     |
    | frontpage   | Web sites that use Microsoft’ s FrontPage server.           |
    | iis         | Web sites that use the Microsoft Internet Information Server (IIS).     |
    | misc        | Miscellaneous attacks.                                       |
    | php         | Web sites that use PHP                                       |
    | web-activex | Web sites that contain ActiveX controls.                     |
    | web-struts  | Web sites that contain Apache struts, which are java-ee based applets. |
-  To display only signatures that have specific check actions enabled, select the ON check box for each of those actions, clear the ON check boxes for the other actions, and clear all of the OFF check boxes. To display only signatures that have a specific check action disabled, select their respective OFF check boxes and clear all of the ON check boxes. To display signatures regardless of whether they have a check action enabled or disabled, select or clear both the ON and the OFF check boxes for that action. The check actions are:
    | Criterion | Description                                                   |
    | --------- | ---------------------------------------------------------------------------
    |
    | Enabled   | The signature is enabled. The App Firewall checks only for signatures that are enabled when it processes traffic.       |
    | Block     | Connections that match this signature are blocked.                    |
    | Log       | A log entry is produced for any connection that matches this signature.                               |
    | Stats     | The App Firewall includes any connection that matches this signature in the statistics that it generates for that check. |
-  To display only signatures that contain a specific string, type the string into the text box under the filter criteria, and then click Search.
-  To reset all display filter criteria to the default settings and display all signatures, click Show All.
```

1. For information about a specific signature, select the signature, and then click the blue double arrow in the More field. The Signature Rule Vulnerability Detail message box appears. It contains information about the purpose of the signature and provides links to external web-based information about the vulnerability or vulnerabilities that this signature addresses. To access an external link, click the blue double arrow to the left of the description of that link.
2. Configure the settings for a signature by selecting the appropriate check boxes.
3. If you want to add a local signature rule to the signatures object, or modify an existing local signature rule, see "The Signatures Editor."
4. If you have no need for SQL injection, cross-site scripting, or Xpath injection patterns, click OK, and then click Close. Otherwise, in the lower left-hand corner of the details pane, click Manage SQL/XSS Patterns.
5. In the Manage SQL/XSS Patterns dialog box, Filtered Results window, navigate to the pattern category and pattern that you want to configure. For information about the SQL injection patterns, see "HTML SQL Injection Check." For information about the cross-site scripting patterns, see "HTML Cross-Site Scripting Check.
6. To add a new pattern:
    a) Select the branch to which you want to add the new pattern.
    b) Click the **Add** button directly below the lower section of the **Filtered Results** window.

   c) In the Create Signature Item dialog box, fill in the Element text box with the pattern that you want to add. If you are adding a transformation pattern to the transform rules branch, under Elements, fill in the From text box with the pattern that you want to change and the To text box with the pattern to which you want to change the previous pattern.
   d) Click **OK**.

7. To modify an existing pattern:

   a) In the **Filtered Results** window, select the branch that contains the pattern that you want to modify.
   b) In the detail window beneath the **Filtered Results** window, select the pattern that you want to modify.
   c) Click **Modify**.
   d) In the **Modify Signature Item** dialog box, **Element** text box, modify the pattern. If you are modifying a transformation pattern, you can modify either or both patterns under Elements, in the From and the To text boxes.
   e) Click **OK**.

8. To remove a pattern, select the pattern that you want to remove, then click the **Remove** button below the details pane beneath the **Filtered Results** window. When prompted, confirm your choice by clicking **Close**.

9. To add the patterns category to the XSS branch:

   a) Select the branch to which you want to add the patterns category.
   b) Click the **Add** button directly below the **Filtered Results** window.

   > **Note**: Currently you can add only one category, named patterns, to the XSS branch, so after you click **Add**,you must accept the default choice, which is patterns.

   3. Click **OK**.

10. To remove a branch, select that branch, and then click the Remove button directly below the **Filtered Results** window. When prompted, confirm your choice by clicking **OK**.

    > **Note**: If you remove a default branch, you remove all of the patterns in that branch. Doing so can disable the security checks that use that information.

11. When you are finished modifying the SQL injection, cross-site scripting, and XPath injection patterns, click **OK**, and then click **Close** to return to the **Modify Signatures Object** dialog box.

12. Click **OK** at any point to save your changes, and when you are finished configuring the signatures object, click **Close**.

Protecting JSON applications using signatures

September 21, 2020

JavaScript Object Notation (JSON) is a text-based open standard derived from the JavaScript scripting language. JSON is preferred for human readable representation of simple data structures and associative arrays, called objects. It serves as an alternative to XML and is primarily used to transmit serialized data structures for communicating with web applications. The JSON files are typically saved with a .json extension.

The JSON payload is typically sent with the MIME type specified as **application/json**. The other "standard" content types for JSON are:

- **application/x-javascript**
- **text/javascript**
- **text/x-javascript**
- **text/x-json**

Using the NetScaler App Firewall signatures to protect JSON applications

To allow JSON requests, the appliance is preconfigured with the JSON content type as shown in the following show-command output:

```
> sh appfw jsonContentType
1)      JSONContenttypevalue:  "^application/json$" IsRegex:  REGEX
Done
<!--NeedCopy-->
```

The NetScaler App Firewall processes the post body for the following content-types only:

- **application/x-www-form-urlencoded**
- **multipart/form-data**
- **text/x-gwt-rpc**

The requests that are received with other content-type headers including application/json (or any other allowed content type) are forwarded to the backend after header inspection. The post body in such requests is not inspected for security check violations even when the profile's security checks such as SQL or XSS are enabled.

In order to protect JSON applications and detect violations, App Firewall signatures can be used. All requests that contain the allowed content-type header are processed by the App Firewall for signature match. You can add your own customized signature rules to process JSON payload to perform various security check inspections (for example, XSS, SQL, and Field Consistency), to detect violations in the headers as well as the post body, and take specified actions.

> **Tip**
>
> Unlike the other built-in defaults, the preconfigured JSON content type can be edited or removed by using the CLI or the GUI (GUI). If legitimate requests for JSON applications are getting blocked and triggering content-type violations, check to make sure that the content type value is configured accurately. For additional details regarding how App Firewall processes content-type header, see Content type protection

To add or remove JSON content-type by using the command line interface

At the command prompt, type one of the following commands:

```
add appfw jsonContentType ^application/json$ IsRegEx REGEX
```

```
rm appfw JSONContentType "^application/json$"
```

To managing JSON content types by using the GUI

Navigate to **Security** > **App Firewall** and, in the **Settings** section, select **Manage JSON Content Types**.

In the **Configure App Firewall JSON Content Type** panel, add, edit, or delete JSON content types to suit the needs of your applications.

Configuring signature protection to detect attacks in JSON payload

In addition to a valid JSON content type, you need to configure signatures to specify the pattern(s) that, when detected in a JSON request, indicate a security breach. The specified actions, such as block and log, are taken when an incoming request triggers a match for all the target patterns in the signature rule.

To add a customized signature rule, Citrix recommends that you use the GUI. Navigate to **System** > **Security** > **App Firewall** > **Signatures**. Double click the target signature object to access the **Edit App Firewall Signatures** panel. Click on the **Add** button to configure the actions, category, log string, rule patterns and so on. Although App Firewall inspects all allowed content-type payload for signature match, you can optimize the processing by specifying the JSON expression in the rule. When you **Add** a new rule pattern, select **Expression** in the drop-down options for **Match** and provide the target match expression from your JSON payload to identify the specific requests that need to be inspected. An expression must begin with a **TEXT.** prefix. You can add other rule patterns to specify additional match patterns to identify the attack.

The following example shows a signature rule. If any cross-site script tag is detected in the POST body of the JSON payload that matches the specified XPATH_JSON expression, a signature match is triggered.

Example of a signature to detect XSS in JSON payload

```
<SignatureRule actions="log,stats" category="JSON" enabled="ON" id="1000001" severity="" source="" type="" version="1">

  <PatternList>

    <RequestPatterns>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="Expression">TEXT.XPATH_JSON(xp%/glossary/title%).CONTAINS("example glossary")</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_METHOD"/>

        <Match type="LITERAL">POST</Match>

      </Pattern>

      <Pattern>

        <Location area="HTTP_POST_BODY"/>

        <Match type="CrossSiteScripting"/>

       </Pattern>

    </RequestPatterns>

  </PatternList>

  <LogString>Cross-site scripting violation detected in json payload</LogString>

  <Comment/>

</SignatureRule>
<!--NeedCopy-->
```

Example of the payload

The following payload triggers the signature match, because it includes the cross-site scripting tag **<Gotcha!!>**.

```
{
 "glossary": {
 "title": "example glossary","GlossDiv": {
 "title": "S","GlossList": {
 "GlossEntry": {
 "ID": "SGML","SortAs": "SGML","GlossTerm": "Standard Generalized Markup Language","Acronym": "SGML","Abbrev": "ISO 8879:1986","GlossDef": {
 "para": "A meta-markup language, used to create markup languages **<Gotcha!!>** such as DocBook.","GlossSeeAlso": ["GML", "XML"] }
 ,"GlossSee": "markup" }
  }
  }
  }
  }

<!--NeedCopy-->
```

Example of the log message

```
Aug 21 12:21:42 <local0.info> 10.217.31.239 08/21/2015:23:21:42 GMT ns 0-PPE-1 : APPFW APPFW_SIGNATURE_MATCH 1471 0 :  10.217.253.62 990-PPE0 NtJnVMNnvPeQJnaUzXYW/GTvAQsA010 prof1 http://10.217.31.212/FFC/login_post.php Signature violation rule ID 1000001: cross-site scripting violation detected in json payload  <not blocked>
<!--NeedCopy-->
```

> **Note**
>
> If you send the same payload after removing the cross-site script tag (**<Gotcha!!>**), the signature rule match is not triggered.

Highlights

- To protect JSON payload, use App Firewall signatures to detect XSS, SQL and other violations.

- Verify that the JSON content type is configured on the appliance as the allowed content type.
- Make sure that the content type in the payload matches the configured JSON content type.
- Make sure that all the patterns configured in the signature rule match for the signature violation to be triggered.
- When you add a signature rule, it MUST have at least one Rule pattern to match the Expression in the JSON payload. All the PI expressions in signature rules must start with the prefix TEXT. and must be Boolean.

Updating a signature object

September 21, 2020

You should update your signatures objects frequently to ensure that your App Firewall is providing protection against current threats. You should regularly update both the default App Firewall signatures and any signatures that you import from a supported vulnerability scanning tool.

Citrix regularly updates the default signatures for the App Firewall. You can update the default signatures manually or automatically. In either case, ask your Citrix representative or Citrix reseller for the URL to access the updates. You can enable automatic updates of the Citrix native format signatures in the “Engine Settings” and “Signature Auto Update Settings” dialog boxes.

Most makers of vulnerability scanning tools regularly update the tools. Most websites also change frequently. You should update your tool and rescan your websites regularly, exporting the resulting signatures to a file and importing them into your App Firewall configuration.

> **Tip**
>
> When you update the App Firewall signatures from the NetScaler command line, you must first update the default signatures, and then issue more update commands to update each custom signatures file that is based on the default signatures. If you do not update the default signatures first, a version mismatch error prevents updating of the custom signatures files.
> **Note**
>
> The following applies to merging a third-party signature object with a user-defined signature object with Native rules and user-added rules:
>
> When a version 0 signatures is merged with a new imported file, the resultant signatures remain as version 0.
>
> This means all native (or built-in) rules in the imported file will be ignored after the merge. This is to ensure that the version 0 signatures are maintained as is after a merge.
>
> To include the native rules in the imported file for merge, you should update the existing signa-

> tures from version 0 first before the merge. This means you need to abandon the version 0 nature of the existing signatures.
>
> When there is a Citrix ADC release upgrade, the file "default_signatures.xml" is added to the new build and the file "updated_signature.xml" is removed from the older build. After the upgrade, if the signature auto update feature is enabled, the appliance updates the existing signature to the latest version of the build and generates the "updated_signature.xml" file.

To update the App Firewall signatures from the source by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergedefault]`
- `save ns config`

Example

The following example updates the signatures object named MySignatures from the default signatures object, merging new signatures in the default signatures object with the existing signatures. This command does not overwrite any user-created signatures or signatures imported from another source, such as an approved vulnerability scanning tool.

```
update appfw signatures MySignatures -mergedefault
save ns config
<!--NeedCopy-->
```

Updating a signatures object from a Citrix format file

Citrix regularly updates the signatures for the App Firewall. You should regularly update the signatures on your App Firewall to ensure that your App Firewall is using the most current list. Ask your Citrix representative or Citrix reseller for the URL to access the updates.

To update a signatures object from a Citrix format file by using the command line

At the command prompt, type the following commands:

- `update appfw signatures <name> [-mergeDefault]`
- `save ns config`

To update a signatures object from a Citrix format file by using the GUI

1. Navigate to **Security** > **App Firewall** > **Signatures**.

2. In the details pane, select the signatures object that you want to update.
3. In the **Action** drop-down list, select **Merge**.
4. In the **Update Signatures Object** dialog box, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local hard drive, network hard drive, or other storage device.
5. In the text area, type the URL, or type or browse to the local file.
6. Click **Update**. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the **Modify Signatures Object** dialog box. The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
7. Review and configure the new and modified signatures.
8. When you are finished, click **OK**, and then click **Close**.

Updating a signatures object from a supported vulnerability scanning tool

> **Note**: Before you update a signatures object from a file, you must create the file by exporting signatures from the vulnerability scanning tool.

To import and update signatures from a vulnerability scanning tool

1. Navigate to **Security** > **App Firewall** > **Signatures**.
2. In the details pane, select the signatures object that you want to update, and then click **Merge**.
3. In the **Update Signatures Object** dialog box, on the **External Format** tab, Import section, choose one of the following options.
   - **Import from URL**—Choose this option if you download signature updates from a Web URL.
   - **Import from Local File**—Choose this option if you import signature updates from a file on your local or a network hard drive or other storage device.
4. In the text area, type the URL, or browse or type the path to the local file.
5. In the XSLT section, choose one of the following options.
   - **Use Built-in XSLT File**—Choose this option if you want to use a built-in XSLT file.
   - **Use Local XSLT File**—Choose this option to use an XSLT file on your local computer.
   - **Reference XSLT from URL**—Choose this option to import an XSLT file from a web URL.
6. If you chose Use Built-in XSLT File, in the Built-In XSLT drop-down list select the file that you want to use from the following options:
   - **Cenzic.**
   - **Deep_Security_for_Web_Apps.**
   - **Hewlett_Packard_Enterprise_WebInspect.**
   - **IBM-AppScan-Enterprise.**

- **IBM-AppScan-Standard.**
- **Qualys.**
- **Whitehat.**

7. Click **Update**. The update file is imported, and the Update Signatures dialog box changes to a format nearly identical to that of the Modify Signatures Object dialog box, which is described in "Configuring or Modifying a Signatures Object." The **Update Signatures Object** dialog box displays all branches with new or modified signature rules, SQL injection or cross-site scripting patterns, and XPath injection patterns if there are any.
8. Review and configure the new and modified signatures.
9. When you are finished, click **OK**, and then click **Close**.

Exporting a signatures object to a file

September 22, 2020

You export a signatures object to a file so that you can import it to another NetScaler.

To export a signatures object to a file

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to configure.
3. In the **Actions** drop-down list, select **Export**.
4. In the **Export Signatures Object** dialog box, **Local File** text box, type the path and name of the file to which you want to export the signatures object, or use the **Browse** dialog to designate a path and name.
5. Click **OK**.

The Signatures editor

September 22, 2020

You can use the signatures editor, which is available in the GUI, to add a new user-defined (local) signature rule to an existing signatures object, or to modify a previously configured local signature rule. Except that it is defined by the user (you), a local signature rule has the same attributes as a default signature rule from Citrix, and it functions in the same way. You enable or disable it, and configure the signature actions for it, just as you do for a default signature.

Add a local rule if you need to protect your web sites and services from a known attack that the existing signatures do not match. For example, you might discover a new type of attack and determine its characteristics by examining the logs on your web server, or you might obtain third-party information about a new type of attack.

At the heart of a signature rule are the rule *patterns*, which collectively describe the characteristics of the attack that the rule is designed to match. Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting patterns.

You might want to modify a signature rule by adding a new pattern or modifying an existing pattern to match an attack. For example, you might find out about changes to an attack, or you might determine a better pattern by examining the logs on your web server, or from third-party information.

To add or modify a local signature rule by using the Signatures Editor

1. Navigate to **Security > Citrix Web App Firewall > Signatures**.
2. In the details pane, select the signatures object that you want to edit, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results** window, do one of the following:
    - To add a new local signature rule, click Add.
    - To modify an existing local signature rule, select that rule, and then click **Open**.
4. In the **Add Local Signature Rule** or the **Modify Local Signature Rule** dialog box, configure the actions for a signature by selecting the appropriate check boxes.
    - **Enabled.** Enables the new signature rule. If you do not select this, this new signature rule is added to your configuration, but is inactive.
    - **Block.** Blocks connections that violate this signature rule.
    - **Log.** Logs violations of this signature rule to the NetScaler log.
    - **Stat.** Includes violations of this signature rule in the statistics.
    - **Remove.** Strips information that matches the signature rule from the response. (Applies only to response rules.)
    - **X-Out.** Masks information that matches the signature rule with the letter X. (Applies only to response rules.)
    - **Allow Duplicates.** Allows duplicates of this signature rule in this signatures object.
5. Choose a category for the new signature rule from the **Category** drop-down list.

    You can also create a new category by clicking the icon to the right of the list and using the Add Signature Rule Category dialog box to add a new category to the list, The rule you are modifying is automatically added to the new category. For instructions, see "To add a signature rule category."
6. In the **LogString** text box, type a brief description of the signature rule to be used in the logs.
7. In the **Comment** text box, type a comment. (Optional)

8. Click More…, and modify the advanced options.
    a) To strip HTML comments before applying this signature rule, in the Strip Comments drop-down list choose All or Exclude Script Tag.
    b) To set CSRF Referer Header checking, in the CSRF Referer Header checking radio button array, select either the If Present or Always radio button.
    c) To manually modify the Rule ID assigned to this local signature rule, modify the number in the Rule ID text box. The ID must be a positive integer between 1000000 and 1999999 that has not already been assigned to a local signature rule.
    d) To assign a version number to the new signature rule, modify the number in the Version Number text box.
    e) To assign a Source ID, modify the string in the Source ID text box.
    f) To specify the source, choose Local or Snort from the Source drop-down list, or click the Add icon to the right of the list and add a new source.
    g) To assign a harm score to violations of this local signature rule, type a number between 1 and 10 in the Harm Score text box.
    h) To assign a severity rating to this local signature rule, in the Severity drop-down list choose High, Medium, or Low, or click the Add icon to the right of the list and add a new severity rating.
    i) To assign a violation type to this local signature rule, in the Type drop-down list choose Vulnerable or Warning, or click the Add icon to the right of the list and add a new violation type.
9. In the **Patterns** list, add or edit a pattern.
    - To add a pattern, click **Add**. In the **Create New Signature Rule Pattern** dialog box, add one or more patterns for your signature rule, and then click **OK**.
    - To edit a pattern, select the pattern, and then click **Open**. In the **Edit Signature Rule Pattern** dialog box, modify the pattern, and then click **OK**.

    For more information about adding or editing patterns, see "Signature Rule Patterns."
10. Click **OK**.

To add a signature rule category

September 21, 2020

Putting signature rules into a category enables you to configure the actions for a group of signatures instead of for each individual signature. You might want to do so for the following reasons:

- **Ease of selection.** For example, assume that all of signature rules in a particular group protect against attacks on a specific type of web server software or technology. If your protected web sites use that software or technology, you want to enable them all. If they do not, you do not want to enable any of them.

- **Ease of initial configuration.** It is easiest to set defaults for a group of signatures as a category, instead of one-by-one. You can then make any changes to individual signatures as needed.
- **Ease of ongoing configuration.** It is easier to configure signatures if you can display only those that meet specific criteria, such as belonging to a specific category.

1. Navigate to **Security** > **App Firewall** > **Signatures**.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen, beneath the **Filtered Results** window, click **Add**.
4. In the **Add Local Signature Rule** dialog box, click the icon to the right of the Category drop-down list.
5. In the **Add Signature Rule Category** dialog box, **New Category** text box, type a name for your new signature category. The name can consist of from one to 64 characters.
6. Click **OK**.

Signature rule patterns

June 9, 2021

You can add a new pattern to a signature rule or modify an existing pattern of a signature rule to specify a string or expression that characterizes an aspect of the attack that the signature matches. To determine which patterns an attack exhibits, you can examine the logs on your web server, use a tool to observe connection data in real time, or obtain the string or expression from a third-party report about the attack.

> **Caution**:
>
> Any new pattern that you add to a signature rule is in an
> AND relationship with the existing patterns. Do not add a new pattern to an existing signature rule if you do not want a potential attack to have to match all of the patterns in order to match the signature.

Each pattern can consist of a simple string, a PCRE-format regular expression, or the built-in SQL injection or cross-site scripting pattern. Before you attempt to add a pattern that is based on a regular expression, you should make sure that you understand PCRE-format regular expressions. PCRE expressions are complex and powerful; if you do not understand how they work, you can unintentionally create a pattern that matches something that you did not want (a *false positive*) or that fails to match something that you did want (a *false negative*).

If you are not already familiar with PCRE-format regular expressions, you can use the following resources to learn the basics, or for help with some specific issue:

- "Mastering Regular Expressions", Third Edition. Copyright (c) 2006 by Jeffrey Friedl. O'Reilly Media, ISBN: 9780596528126.
- "Regular Expressions Cookbook". Copyright (c) 2009 by Jan Goyvaerts and Steven Levithan. O'Reilly Media, ISBN: 9780596520687
- **PCRE Man page/Specification** (text/official): http://www.pcre.org/pcre.txt
- **PCRE Man Page/Specification**

http://www.gammon.com.au/pcre/index.html

- **Wikipedia PCRE entry**: http://en.wikipedia.org/wiki/PCRE
- **PCRE Mailing List**
  Pcre-dev - PCRE Development

If you need to encode non-ASCII characters in a PCRE-format regular expression, the NetScaler platform supports encoding of hexadecimal UTF-8 codes. For more information, see "PCRE Character Encoding Format."

To configure a signature rule pattern

1. Navigate to **Security > Web App Firewall > Signatures**.
2. In the details pane, select that signatures object that you want to configure, and then click **Open**.
3. In the **Modify Signatures Object** dialog box, in the middle of the screen beneath the **Filtered Results** window, either click **Add** to create a signature rule, or select an existing signature rule and click **Open**.

   > **Note**: You can modify only signature rules that you added. You cannot modify the default signature rules.

   Depending on your action, either the Add Local Signature Rule or the Modify Local Signature Rule dialog box appears. Both dialog boxes have the same contents.
4. Under the **Patterns window** in the dialog box, either click **Add** to add a new pattern, or select an existing pattern from the list beneath the **Add** button and click **Open**. Depending on your action, either the **Create New Signature Rule Pattern** or the **Edit Signature Rule Pattern** dialog box appears. Both dialog boxes have the same contents.
5. In the **Pattern Type** drop-down list, choose the type of connection that the pattern is intended to match.
   - If the pattern is intended to match request elements or features, such as injected SQL code, attacks on web forms, cross-site scripts, or inappropriate URLs, choose **Request**.

- If the pattern is intended to match response elements or features, such as credit card numbers or safe objects, choose **Response**.

6. In the Location area, define the elements to examine with this pattern.

   The Location area describes what elements of the HTTP request or response to examine for this pattern. The choices that appear in the Location area depend upon the chosen pattern type. If you chose Request as the pattern type, items relevant to HTTP requests appear; if you chose Response, items relevant to HTTP responses appear.

   In addition, as you choose a value from the Area drop-down list, the remaining parts of the Location area change interactively. Following are all configuration items that might appear in this section.

   - Area Drop-down list of elements that describe a particular portion of the HTTP connection. The choices are as follows:
     - **HTTP_ANY**. All parts of the HTTP connection.
     - **HTTP_COOKIE**. All cookies in the HTTP request headers after any cookie transformations are performed.
       Note: Does not search HTTP response "Set-Cookie:" headers.
     - **HTTP_FORM_FIELD**. Form fields and their contents, after URL decoding, percent decoding, and removal of excess whitespace. You can use the `<Location>` tag to further restrict the list of form field names to be searched.
     - **HTTP_HEADER**. The value portions of the HTTP header after any cross-site scripting or URL decoding transformations.
     - **HTTP_METHOD**. The HTTP request method.
     - **HTTP_ORIGIN_URL**. The origin URL of a web form.
     - **HTTP_POST_BODY**. The HTTP post body and the web form data that it contains.
     - **HTTP_RAW_COOKIE**. All HTTP request cookie, including the "Cookie:" name portion.
       Note: Does not search HTTP response "Set-Cookie:" headers.
     - **HTTP_RAW_HEADER**. The entire HTTP header, with individual headers separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
     - **HTTP_RAW_RESP_HEADER**. The entire response header, including the name and value parts of the response header after URL transformation has been done, and the complete response status. As with HTTP_RAW_HEADER, individual headers are separated by linefeed characters (\n) or carriage return/line-feed strings (\r\n).
     - **HTTP_RAW_SET_COOKIE**. The entire Set-Cookie header after any URL transformations have been performed
       Note: URL transformation can change both the domain and path parts of the Set-Cookie header.
     - **HTTP_RAW_URL**. The entire request URL before any URL transformations are performed, including any query or fragment parts.

    - **HTTP_RESP_HEADER**. The value part of the complete response headers after any URL transformations have been performed.
    - **HTTP_RESP_BODY**. The HTTP response body
    - **HTTP_SET_COOKIE**. All "Set-Cookie" headers in the HTTP response headers.
    - **HTTP_STATUS_CODE**. The HTTP status code.
    - **HTTP_STATUS_MESSAGE**. The HTTP status message.
    - **HTTP_URL**. The value portion of the URL in the HTTP headers, excluding any query or fragment ports, after conversion to the UTF-* character set, URL decoding, stripping of whitespace, and conversion of relative URLs to absolute. Does not include HTML entity decoding.
  - URL
    Examines any URLs found in the elements specified by the
    Area setting. Select one of the following settingss.
    - **Any.** Checks all URLs.
    - **Literal**. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
    - **PCRE**. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
    - **Expression**. Checks URLs that match a NetScaler default expression.
  - **Field Name**. Examines any form field names found in the elements specified by the Area selection.
  - **Any.** Checks all URLs.
    - Literal. Checks URLs that contain a literal string. After you select Literal, a text box is displayed. Type the literal string that you want in the text box.
    - PCRE. Checks URLs that match a PCRE-format regular expression. After you select this choice, the regular expression window is displayed. Type the regular expression in the window. You can use the **Regex Tokens** to insert common regular expression elements at the cursor, or you can click Regex Editor to display the Regular Expression Editor dialog box, which provides more assistance in constructing the regular expression that you want.
    - Expression. Checks URLs that match a NetScaler default expression.

7. In the Pattern area, define the pattern. A pattern is a literal string or PCRE-format regular expression that defines the pattern that you want to match. The Pattern area contains the following elements:

   - Match

A drop-down list of search methods that you can use for the signature. This list differs depending on whether the pattern type is Request or Response.

Request Match Types

PCRE. A PCRE-format regular expression.

NOTE: When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Injection**. Directs the App Firewall to look for injected SQL in the specified location. The Pattern window disappears, because the App Firewall already has the patterns for SQL injection.
- **CrossSiteScripting**. Directs the App Firewall to look for cross-site scripts in the specified location. The Pattern window disappears, because the App Firewall already has the patterns for cross-site scripts.
- **Expression**. An expression in the NetScaler default expressions language. This is the same expressions language that is used to create App Firewall policies and other policies on the NetScaler appliance. Although the NetScaler expressions language was originally developed for policy rules, it is a highly flexible general purpose language that can also be used to define a signature pattern.

When you choose Expression, the NetScaler Expression Editor appears beneath Pattern window. For more information about the Expression Editor and instructions on how to use it, see "To add a firewall rule (expression) by using the Add Expression dialog box

Response Match Types:

- **Literal**. A literal string
- **PCRE**. A PCRE-format regular expression.

> **NOTE:**
>
> When you choose PCRE, the regular expression tools beneath the Pattern window are enabled. These tools are not useful for most other types of patterns.

- **Credit Card**. A built-in pattern to match one of the six supported types of credit card number.

Note: The Expression match type is not available for Response-side signatures.

- Pattern Window (unlabeled)

In this window, type the pattern that you want to match, and fill in any additional data.

```
-  Literal. Type the string you want to search for in the text area.
-  PCRE. Type the regular expression in the text area. Use the **Regex
    Editor** for more assistance in constructing the regular expression
    that you want, or the Regex Tokens to insert common regular
    expression elements at the cursor. To enable UTF-8 characters, click
     UTF-8.
```

```
- Expression. Type the NetScaler advanced expression in the text area.  Use Prefix to choose the first term in your expression, or Operator  to insert common operators at the cursor. Click **Add** to open the  Add Expression dialog box for more assistance in constructing the regular expression that you want. Click Evaluate to open the Advanced Expression Evaluator to help determine what effect your expression has.
- Offset. The number of characters to skip over before starting to match on this pattern. You use this field to start examining a string at some point other than the first character.
- Depth. How many characters from the starting point to examine for matches. You use this field to limit searches of a large string to a  specific number of characters.
- Min-Length. The string to be searched must be at least the specified  number of bytes in length. Shorter strings are not matched.
- Max-Length. The string to be searched must be no longer than the specified number of bytes in length. Longer strings are not matched.
- Search method. A check box labeled fastmatch. You can enable fastmatch only for a literal pattern, to improve performance.
```

1. Click **OK**.
2. Repeat the previous four steps to add or modify additional patterns.
3. When finished adding or modifying patterns, click **OK** to save your changes and return to the Signatures pane.

   > **Caution**:
   >
   > Until you click **OK** in the **Add Local Signature Rule** or **Modify Local Signature Rule** dialog box, your changes are not saved. Do not close either of these dialog boxes without clicking **OK** unless you want to discard your changes.

To Import and merge rules

September 21, 2020

When using the signature editor to perform an import and merge operation from the GUI, you can now see the new, updated, duplicate, and invalid rules.

The signature editor displays the following four new rows:

1. New Rules
2. Updated Rules

3. Duplicate Rules
4. Invalid Rules

The output of the New Rules Only and Updated Rules Only filters also appears in the Category filter pane of the Edit window in signature editor.

You will need to import the files from GUI to see the corresponding links for New, duplicate, invalid and updated rules.

For example, you can use GUI to import the following signature files:

`<http://10.217.30.16/testsite/signatures/sig-3100000.xml>`.

To import signature rules:

1. In the NetScaler web GUI, go to **Configuration** > **Security** > **Application Firewall** > **Signatures**. In the Signatures window, click **Add**. Then select **File Format** > **Native**, **Import From** > **URL** and in the URL field, add the above link. For example; `<http://10.217.30.16/testsite/signatures/sig-3100000.xml>`.
2. After you click **Open**, the signature file will open and you can see links for New Rule and Invalid Rules.
3. If you import a `3<sup>rd</sup>` party signature rule from the following site, for example; `<http://10.217.30.16/FFC/sig_validation/trendmicro_sample2.xml>` as shown below, you can see 90 new Rules and 9 duplicate Rules in the imported .xml file.

Signature updates in high availability deployment and build upgrades

September 21, 2020

The signature update occurs on the primary node. While the signatures are updated on the primary node, in parallel the updated files are simultaneously synchronized with the secondary node.

The *Default signature is always updated first and then the rest of the user-defined signatures are updated.

Connecting to Amazon AWS

The default route NSIP is used to connect to the Amazon AWS. If there is a specific use case scenario where SNIP is used, and if there are multiple SNIPs, the first one to receive the ARP response from the hosting site will hold the route.

Signature updates during version upgrades

In case of an upgrade, if the NS has an older base version for the signatures, *Default signature is automatically updated if a newer signature version is available.

If the schema has changed, the schema version of all the signature objects gets updated when the version is upgraded.

However, for the base version of the user-defined signatures, the behavior is different in release 10.5 versus release 11.0.

In release 10.5, only the default signature was updated and the base version of the rest of the signatures remained unchanged after the build upgrade.

In release 11.0, this behavior has changed. When the appliance is upgraded to install a new build, not only the *Default signature object but all the other user-defined signatures that currently exist in the appliance are also updated and will have the same version after the build upgrade.

In both 10.5 and 11.0 release builds, if auto-update is configured, the *Default Signatures as well as all non-zero version signatures get auto-updated to the latest released signature version and will have the same base version.

Overview of security checks

September 21, 2020

The App Firewall advanced protections (security checks) are a set of filters designed to catch complex or unknown attacks on your protected web sites and web services. The security checks use heuristics, positive security, and other techniques to detect attacks that may not be detected by signatures alone. You configure the security checks by creating and configuring an App Firewall profile, which is a collection of user-defined settings that tell the App Firewall which security checks to use and how to handle a request or response that fails a security check. A profile is associated with a signatures object and with a policy to create a security configuration.

The App Firewall provides twenty security checks, which differ widely in the types of attacks that they target and how complex they are to configure. The security checks are organized into the following categories:

- **Common security checks.** Checks that apply to any aspect of web security that either does not involve content or is equally applicable to all types of content.
- **HTML security checks.** Checks that examine HTML requests and responses. These checks apply to HTML-based web sites and to the HTML portions of Web 2.0 sites, which contain mixed HTML and XML content.

- **XML security checks.** Checks that examine XML requests and responses. These checks apply to XML-based web services and to the XML portions of Web 2.0 sites.

The security checks protect against a wide range of types of attack, including attacks on operation system and web server software vulnerabilities, SQL database vulnerabilities, errors in the design and coding of web sites and web services, and failures to secure sites that host or can access sensitive information.

All security checks have a set of configuration options, the check actions, which control how the App Firewall handles a connection that matches a check. Three check actions are available for all security checks. They are:

- **Block.** Block connections that match the signature. Disabled by default.
- **Log.** Log connections that match the signature, for later analysis. Enabled by default.
- **Stats.** Maintain statistics, for each signature, that show how many connections it matched and provide certain other information about the types of connections that were blocked. Disabled by default.

A fourth check action, **Learn**, is available for more than half of the check actions. It observes traffic to a protected Web site or web service and uses connections that repeatedly violate the security check to generate recommended exceptions (relaxations) to the check, or new rules for the check. In addition to the check actions, certain security checks have parameters that control the rules that the check uses to determine which connections violate that check, or that configure the App Firewall's response to connections that violate the check. These parameters are different for each check, and they are described in the documentation for each check.

To configure security checks, you can use the App Firewall wizard, as described in "The App Firewall Wizard," or you can configure the security checks manually, as described in "Manual Configuration By Using the GUI." Some tasks, such as manually entering relaxations or rules or reviewing learned data, can be done only by using the GUI, not the command line. Using the wizard is usually best configuration method, but in some cases manual configuration might be easier if you are thoroughly familiar with it and simply want to adjust the configuration for a single security check.

Regardless of which method you use to configure the security checks, each security check requires that certain tasks be performed. Many checks require that you specify exceptions (relaxations) to prevent blocking of legitimate traffic before you enable blocking for that security check. You can do this manually, by observing the log entries after a certain amount of traffic has been filtered and then creating the necessary exceptions. However, it is usually much easier to enable the learning feature and let it observe the traffic and recommend the necessary exceptions.

App Firewall uses packet engines (PE) during processing the transactions. Each packet engine has a limit of 100K sessions which is sufficient for most deployment scenarios. However, when App Firewall is processing heavy traffic and the session timeout is configured at a higher value, the sessions might get accumulated. If the number of alive App Firewall sessions exceed the 100K per PE limit, the App Firewall security check violations might not be sent to the Security Insight appliance. Lowering the session timeout to a smaller value, or using sessionless mode for the security checks with sessionless URL closure or sessionless field consistency might help in preventing the sessions getting accumulated. If this is not a viable option in scenarios where transactions might require longer sessions, upgrading to a higher-end platform with more packet engine is recommended.

Support for cached AppFirewall is added, and the max session setting through the CLI per core is set to 50K sessions.

Top level protections

September 21, 2020

Four of the App Firewall protections are especially effective against common types of Web attacks, and are therefore more commonly used than any of the others. They are:

- **HTML Cross-Site Scripting.** Examines requests and responses for scripts that attempt to access or modify content on a different Web site than the one on which the script is located. When this check finds such a script, it either renders the script harmless before forwarding the request or response to its destination, or it blocks the connection.
- **HTML SQL Injection.** Examines requests that contain form field data for attempts to inject SQL commands into an SQL database. When this check detects injected SQL code, it either blocks the request or renders the injected SQL code harmless before forwarding the request to the Web server.

  Note: If both of the following conditions apply to your configuration, you should make certain that your App Firewall is correctly configured:

  - If you enable the HTML Cross-Site Scripting check or the HTML SQL Injection check (or both), and
  - Your protected Web sites accept file uploads or contain Web forms that can contain large POST body data.

  For more information about configuring the App Firewall to handle this case, see "Configuring the Application Firewall."
- **Buffer Overflow.** Examines requests to detect attempts to cause a buffer overflow on the Web server.
- **Cookie Consistency.** Examines cookies returned with user requests to verify that they match the cookies your Web server set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the Web server.

The Buffer Overflow check is simple; you can usually enable blocking for it immediately. The other three top-level checks are considerably more complex and require configuration before you can safely use them to block traffic. Citrix strongly recommends that, rather than attempting to configure these checks manually, you enable the learning feature and allow it to generate the necessary exceptions.

HTML cross-site scripting check

September 21, 2020

The HTML Cross-Site Scripting (XSS) check examines both the headers and the POST bodies of user requests for possible cross-site scripting attacks. If it finds a cross-site script, it either modifies (*transforms*) the request to render the attack harmless, or blocks the request.

> **Note:**
>
> The HTML Cross-Site Scripting (XSS) check works only for content type, content length, etc. It does not work for cookie. Also ensure to have the 'checkRequestHeaders' option enabled in your Web Application Firewall profile.

To prevent misuse of the scripts on your protected websites to breach security on your websites, the HTML Cross-Site Scripting check blocks scripts that violate the *same origin rule*, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

Unfortunately, many companies have a large installed base of JavaScript-enhanced web content that violates the same origin rule. If you enable the HTML Cross-Site Scripting check on such a site, you have to generate the appropriate exceptions so that the check does not block legitimate activity.

The App Firewall offers various action options for implementing HTML Cross-Site Scripting protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, you also have the option to **Transform cross-site scripts** to render an attack harmless by entity encoding the script tags in the submitted request. You can configure Check complete URLs for cross-site scripting parameter to specify if you want to inspect not just the query parameters but the entire URL to detect XSS attack. You can configure **Check Request Query Non HTML** parameter to inspect request query portion for XSS attack irrespective of content-type.

You can deploy relaxations to avoid false positives. The App Firewall learning engine can provide recommendations for configuring relaxation rules.

Following options are available for configuring an optimized HTML Cross-Site Scripting protection for your application:

- **Block**—If you enable block, the block action is triggered if the XSS tags are detected in the request.
- **Log**—If you enable the log feature, the HTML Cross-Site Scripting check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each header or form field in which the XSS violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when XSS tags are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
- **Stats**—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you must configure new relaxation rules or modify the existing ones.
- **Learn**—If you are not sure which relaxation rules might be ideally suited for your application, you can use the learn feature to generate HTML Cross-Site Scripting rule recommendations based on the learned data. The App Firewall learning engine monitors the traffic and provides learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
- **Transform cross-site scripts**—If enabled, the App Firewall makes the following changes to requests that match the HTML Cross-Site Scripting check:
  - Left angle bracket (<) to HTML character entity equivalent (<)
  - Right angle bracket (>) to HTML character entity equivalent (>)

This ensures that browsers do not interpret unsafe html tags, such as <`script`>, and thereby execute malicious code. If you enable both request-header checking and transformation, any special characters found in request headers are also modified as described above. If scripts on your protected website contain cross-site scripting features, but your website does not rely upon those scripts to operate correctly, you can safely disable blocking and enable transformation. This configuration ensures that no legitimate web traffic is blocked, while stopping any potential cross-site scripting attacks.

- **Check complete URLs for cross-site scripting**—If checking of complete URLs is enabled, the App Firewall examines entire URLs for HTML cross-site scripting attacks instead of checking just the query portions of URLs.
- **Check Request headers**—If Request header checking is enabled, the App Firewall examines the headers of requests for HTML cross-site scripting attacks, instead of just URLs. If you use the GUI, you can enable this parameter in the Settings tab of the App Firewall profile.
- **Check Request Query for Non HTML** — If Request query checking is enabled, the App Firewall examines the query of requests for cross-site scripting attacks irrespective of content-type. If

you use the GUI, you can enable this parameter in the Settings tab of the App Firewall profile.

> **Important**
>
> As part of the streaming changes, the App Firewall processing of the Cross-site Scripting tags has changed. This change is applicable to 11.0 builds onwards. This change is also pertinent for the enhancement builds of 10.5.e that support request side streaming. In earlier releases, presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as Cross-site Scripting Violation. The behavior has changed in the builds that include support for request side streaming. Only the close bracket character (>) is no longer considered as an attack. Requests are blocked even when an open bracket character (<) is present, and is considered as an attack. The Cross-site scripting attack gets flagged.

XSS Fine grained Relaxations

The App Firewall gives you an option to exempt a specific form field, header, or Cookie from cross-site scripting inspection check. You can completely bypass the inspection for one or more of these fields by configuring relaxation rules.

The App Firewall allows you to implement tighter security by fine-tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any cross-site scripting attack patterns. Cross-site scripting fine grained relaxation provides the option to allow specific attributes, tags, and patterns. The rest of the attributes, tags and patterns are blocked. For example, the App Firewall currently has a default set of more than 125 denied patterns. Because hackers can use these patterns in Cross-site script attacks, the App Firewall flags them as potential threats. You can relax one or more patterns that are considered safe for the specific location. The rest of the potentially dangerous XSS patterns are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. The value type can be left blank or you have an option to select **Tag** or **Attribute** or **Pattern**. If you leave the value type blank, the configured field of the specified URL is exempted from the Cross-Site Scripting check inspection. If you select a value type, you must provide a value expression. You can specify whether the value expression is a regular expression or a literal string. When the input is matched against the allowed and denied list, only the specified expressions configured in the relaxation rules are exempted.

The App Firewall has the following XSS built-in lists:

1. **XSS Allowed Attributes**: There are 52 default allowed attributes, such as, **abbr**, **accesskey**, **align**, **alt**, **axis**, **bgcolor**, **border**, **cellpadding**, **cell spacing**, **char**, **charoff**, **charset** and so forth

2. **XSS Allowed Tags**: There are 47 default allowed tags, such as, **address**, **basefont**, **bgsound**, **significant**, **blockquote**, **bg**, **br**, **caption**, **center**, **cite**, **dd**, **del** and so forth
3. **XSS Denied Patterns**: There are 129 default denied patterns, such as, **FSCommand**, **javascript:**, **onAbort**, **onActivate** and so forth

> **Warning**
>
> App Firewall action URLs are regular expressions. When configuring HTML cross-site scripting relaxation rules, you can specify **Name**, and **Value Expression** to be literal or RegEx. Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the rule that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intends to block or allowing an attack that the HTML Cross-Site Scripting check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.
- Value expressions should be assigned a value type. The XSS value type can be: 1) Tag, 2) Attribute, or 3) Pattern.
- You can have multiple relaxation rules per field name/URL combination
- The form field names and the action URLs are not case sensitive.

Using the Command Line to Configure the HTML Cross-Site Scripting check

To configure HTML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the HTML Cross-Site Scripting Check:

- set appfw profile "Parameter descriptions provided at bottom of the page.")
- `<name> -crossSiteScriptingAction (([block] [learn] [log] [stats])| [**none**])`
- set appfw profile "Parameter descriptions provided at bottom of the page.")
- `<name> **-crossSiteScriptingTransformUnsafeHTML** (ON | OFF)`
- set appfw profile Parameter descriptions provided at bottom of the page.
- `<name> -crossSiteScriptingCheckCompleteURLs (ON | OFF)`
- set appfw profile Parameter descriptions provided at bottom of the page.
- `<name> -checkRequestHeaders (ON | OFF)` Parameter descriptions provided at bottom of the page.

- `<name> -CheckRequestQueryNonHtml (ON | OFF)` Parameter descriptions provided at bottom of the page.

To configure an HTML Cross-Site Scripting check relaxation rule by using the command line

Use the bind or unbind command to add or delete binding, as follows:

- `bind appfw profile <name> -crossSiteScripting <String> [isRegex (REGEX | NOTREGEX)] <formActionURL> [-location <location>] [-valueType (Tag|Attribute|Pattern)[<valueExpression>] [-isValueRegex (REGEX | NOTREGEX)]]`
- `unbind appfw profile <name> -crossSiteScripting <String> <formActionURL> [-location <location>] [-valueType (Tag |Attribute|Pattern)[<valueExpression>]]`

Using the GUI to configure the HTML Cross-Site scripting check

In the GUI, you can configure the HTML Cross-Site Scripting check in the pane for the profile associated with your application.

To configure or modify the HTML Cross-Site Scripting check by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for the HTML Cross-Site Scripting, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

b. If you want to configure additional options for this security check, double-click **HTML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the following options:

Transform cross-site scripts—Transform unsafe script tags.

Check complete URLs for Cross-site scripting—Instead of checking just the query part of the URL, check the complete URL for cross-site script violations.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the **Security Check** pane.

To enable or disable the **Check request Header** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK**. You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **Application Firewall** > **Profile**.

To enable or disable the **Check request Query Non HTML** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check request Query Non HTML** check box. Click **OK**. You can either use the X icon at the top right hand side of the **Profile Settings** pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **App Firewall > Profile**.

To configure an HTML Cross-Site Scripting relaxation rule by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **HTML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **HTML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Tag** or **Attribute** or **Pattern** option in the **Value Type** Field.

To manage HTML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the GUI

You can use the GUI to view or customize the default list of XSS allowed attributes or allowed tags. You can also view or customize the default list of XSS denied Patterns.

The default lists are specified in **Application Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default XSS allowed and denied list specified in the Default Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the allowed or denied lists in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized allowed and denied lists.

For more information about Signatures, see the following topic..

1. To view default XSS patterns:

a. Navigate to **Application Firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

The M**anage SQL/XSS Paths** table shows following three rows pertaining to XSS :

`xss/allowed/attribute`

`xss/allowed/tag`

`xss/denied/pattern`

b. Select a row and click **Manage Elements** to display the corresponding XSS Elements (Tag, Attribute, Pattern) used by the App Firewall **Cross-Site Scripting** check.

1. **To customize XSS Elements**: You can edit the User-Defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

a. Navigate to **Application Firewall** > **Signatures**, highlight the target User-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.

b. Select the target XSS row.

i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding XSS element.

ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful before you remove or modify any default XSS element, or delete the XSS path to remove the entire row. The signature rules as well as the Cross-Site Scripting security check rely on these elements for detecting attacks to protect your applications. Customizing the XSS Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the Learn Feature with the HTML Cross-Site Scripting Check

When the learn action is enabled, the App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as an HTML Cross-Site Scripting relaxation rule.

HTML Cross-Site Scripting Learning enhancement—An App Firewall learning enhancement was introduced in release 11.0 of the NetScaler software. To deploy fine grained HTML Cross-Site Scripting relaxation, the App Firewall offers fine grained HTML Cross-Site Scripting learning. The learning engine makes recommendations regarding the observed Value Type (Tag, Attribute, Pattern) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in relaxation rules.

> **Note**
>
> The App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from HTML Cross-Site Scripting inspection.
> **Tip**
>
> XSS tags which are longer than 12 characters are not learned or logged correctly.
>
> If you need a larger tag length for learning, you can add a large non-appearing tag in the **AS_XSS_ALLOWED_TAGS_LIST** for length 'x'.

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- `show appfw learningdata <profilename> crossSiteScripting`
- `rm appfw learningdata <profilename> -crossSiteScripting <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> **crossSiteScripting*`

To view or use learned data by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML Cross-Site Scripting** entry in the Learned Rules table and double-click it to access the learned rules. The table displays the **Field Name**, A**ction URL**, **Value Type**, **Value, and **Hits** columns. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the **HTML Cross-Site Scripting** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of all the learned violations. The visualizer makes it easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the Log Feature with the HTML Cross-Site Scripting Check

When the log action is enabled, the HTML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_XSS** violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the HTML Cross-Site Scripting violations:

> ```
> Shell
> tail -f /var/log/ns.log | grep APPFW_XSS
> ```

Example of a Cross-Site Scripting security check violation log message in CEF log format

```
Jul 11 00:45:51 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|**APPFW_XSS**|6|src=10.217.253.62 geolocation=Unknown spt
    =4840 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html?abc\=%3Cdef%3E msg=**Cross-site script check
    failed for field abc=\"Bad tag: def"** cn1=133 cn2=294 cs1=pr_ffc
    cs2=PPE1 cs3=eUljypvLa0BbabwfGVE52Sewg9U0001 cs4=ALERT cs5=2015 act
    =**not blocked**
<!--NeedCopy-->
```

Example of a Cross-Site Scripting security check violation log message in Native log format showing transform action

```
Jul 11 01:00:28 <local0.info> 10.217.31.98 07/11/2015:01:00:28 GMT ns
    0-PPE-0 : default APPFW **APPFW_XSS** 132 0 :  10.217.253.62 392-
    PPE0 eUljypvLa0BbabwfGVE52Sewg9U0001 pr_ffc http://aaron.stratum8.
    net/FFC/login.php?login_name=%3CBOB%3E&passwd=&drinking_pref=on &
    text_area=&loginButton=ClickToLogin&as_sfid=
    AAAAAAVFqmYL68IGvkrcn2pzehjfIkm5E6EZ9FL8YLvIW_41AvAATuKYe9N7uGThSpEAxbb0iBx55j
    -FC4llF **Cross-site script special characters seen in fields <
    transformed>**
<!--NeedCopy-->
```

To access the log messages by using the GUI

The NetScaler GUI includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML Cross-Site Scripting** row and click **Logs**. When you access the logs directly from the HTML Cross-Site Scripting check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.

- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML Cross-Site Scripting** check, filter by selecting **APPFW** in the drop-down list options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XSS** check box and click the **Apply** button, only log messages pertaining to the HTML Cross-Site Scripting security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** and so forth appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy HTML Cross-Site Scripting relaxation rules based on the log messages for the App Firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the **Syslog Viewer** box of the selected row. Select the check box, and then select an option from the **Action** list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The HTML Cross-Site Scripting rules that are deployed by using the **Click to Deploy** option do not include the fine grain relaxation recommendations.

To use Click to Deploy functionality in the GUI

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Select the **APP_XSS** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the HTML Cross-Site Scripting violations

When the stats action is enabled, the counter for the HTML Cross-Site Scripting check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 HTML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked when the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display HTML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> **stat appfw profile** <profile name>
```

To display HTML Cross-Site Scripting statistics by using the GUI

1. Navigate to **Security** > **Application Firewall > Profiles > Statistics**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about HTML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the HTML Cross-Site Scripting check:

- **Built-in Support for HTML Cross-Site Scripting attack Protection**—The NetScaler App Firewall protects against Cross-Site Scripting attacks by monitoring a combination of allowed attributes and tags, as well as denied patterns in the received payload. All the built-in default allowed tags, allowed attributes and denied patterns used by the XSS check are specified in the /netscaler/default_custom_settings.xml file.
- **Customizatio**n—You can change the default list of tags, attributes, and patterns to customize the Cross-Site Scripting security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/XSS patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/XSS patterns present in the default signature object are used.
- **Transform**—Note the following about the transform operation:

The transform operation works independently of the other Cross-Site Scripting action settings. If transform is enabled and block, log, stats, and learn are all disabled, XSS tags are transformed.

If the block action is enabled, it takes precedence over the transform action.

- **Fine Grained Relaxation and Learning**—Fine-tune the relaxation rule to relax a subset of XSS elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deplo**y—Select one, or multiple XSS violation log messages in the syslog viewer and deploy them as relaxation rules.
- **Charset**—The default charset for the profile should be set based on the need of the application. By default, the profile charset is set to English US (ISO-8859-1). If a request is received without the specified charset, the App Firewall processes the request as if it is ISO-8859-1. The open bracket character (<) or the close bracket character (>) will not get interpreted as XSS tags if these characters are encoded in other charsets. For example, if a request contains a UTF-8 character string "**%uff1cscript%uff1e**" but the charset is not specified on the request page, the XSS violation might not get triggered unless the default charset for the profile is specified as Unicode.

HTML SQL injection check

September 21, 2020

Many web applications have web forms that use SQL to communicate with relational database servers. Malicious code or a hacker can use an insecure web form to send SQL commands to the web server. The App Firewall HTML SQL Injection check provides special defenses against injection of unauthorized SQL code that might break security. If the App Firewall detects unauthorized SQL code in a user request, it either transforms the request, to render the SQL code inactive, or blocks the request. The App Firewall examines the request payload for injected SQL code in three locations: 1) POST body, 2) headers, and 3) cookies. In order to examine query portion in requests for injected SQL code irrespective of content-type, please enable appfw profile setting 'Check Request Query Non HTML'.

A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch SQL attacks. You can add new patterns, and you can edit the default set to customize the SQL check inspection. The App Firewall offers various action options for implementing SQL Injection protection. In addition to the **Block**, **Log**, **Stats** and **Learn** actions, the App Firewall profile also offers the option to **transform SQL special characters** to render an attack harmless.

In addition to actions, there are several parameters that can be configured for SQL injection processing. You can check for **SQL wildcard characters**. You can change the SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the payload. The **SQL Comments Handling parameter** gives you an option to specify the type of comments that need to be inspected or exempted during SQL Injection detection.

You can deploy relaxations to avoid false positives. The App Firewall learning engine can provide recommendations for configuring relaxation rules.

Following options are available for configuring an optimized SQL Injection protection for your application:

Block—If you enable block, the block action is triggered only if the input matches the SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the SQL injection type, a request is not blocked if it contains no key words, even if SQL special characters are detected in the input. Such a request is blocked if the SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

Log—If you enable the log feature, the SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each input field in which the SQL violation was detected. However, only one message is generated when the request is blocked. Similarly, one log message per request is generated for the transform operation, even when SQL special characters are transformed in multiple fields. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Learn—If you are not sure which SQL relaxation rules might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The App Firewall learning engine monitors the traffic and provides SQL learning recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.

Transform SQL special characters—The App Firewall considers three characters, Single straight quote (‘), Backslash (), and Semicolon (;) as special characters for SQL security check processing. The SQL Transformation feature modifies the SQL Injection code in an HTML request to ensure that the request is rendered harmless. The modified HTML request is then sent to the server. All default transformation rules are specified in the /netscaler/default_custom_settings.xml file.

The transform operation renders the SQL code inactive by making the following changes to the request:

- Single straight quote (‘) to double straight quote (“).
- Backslash () to double backslash ().
- Semicolon (;) is dropped completely.

These three characters (special strings) are necessary to issue commands to an SQL server. Unless an SQL command is prefaced with a special string, most SQL servers ignore that command. Therefore, the changes that the App Firewall performs when transformation is enabled prevent an attacker from injecting active SQL. After these changes are made, the request can safely be forwarded to your protected web site. When web forms on your protected web site can legitimately contain SQL special strings, but the web forms do not rely on the special strings to operate correctly, you can disable blocking and enable transformation to prevent blocking of legitimate web form data without reducing the protection that the App Firewall provides to your protected web sites.

The transform operation works independently of the SQL Injection Type setting. If transform is enabled and the SQL Injection type is specified as SQL keyword, SQL special characters are transformed even if the request does not contain any keywords.

> **Tip**
>
> You normally enable either transformation or blocking, but not both. If the block action is enabled, it takes precedence over the transform action. If you have blocking enabled, enabling transformation is redundant.

Check for SQL Wildcard Characters—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

SELECT * from customer WHERE name like “%D%”:

The following example combines the operators to find any salary values that have 0 in the second and third place.

SELECT * from customer WHERE salary like ‘_00%’:

Different DBMS vendors have extended the wildcard characters by adding extra operators. The NetScaler App App Firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

- `<wildchar type=" LITERAL" >%</wildchar>`
- `<wildchar type=" LITERAL" >_</wildchar>`
- `<wildchar type=" LITERAL" >^</wildchar>`
- `<wildchar type=" LITERAL" >[</wildchar>`
- `<wildchar type=" LITERAL" >]</wildchar>`

Wildcard characters in an attack can be PCRE, like [^A-F]. The App Firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note**
>
> The SQL wildcard character check is different from the SQL special character check. This option must be used with caution to avoid false positives.

Check Request Containing SQL Injection Type—The App Firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the input to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the input to trigger SQL violation.
- **SQL key word**—At least one of the specified SQL keywords must be present in the input to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.
- **SQL Special Character or Keyword**—Either the key word or the special character string must be present in the input to trigger the security check violation.

> **Tip**
>
> If you configure the App Firewall to check for inputs that contain an SQL special character, the App Firewall skips web form fields that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the App Firewall and speed up processing without placing your protected web sites at risk.

SQL comments handling—By default, the App Firewall checks all SQL comments for injected SQL commands. Many SQL servers ignore anything in a comment, however, even if preceded by an SQL special character. For faster processing, if your SQL server ignores comments, you can configure the

App Firewall to skip comments when examining requests for injected SQL. The SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases. For example:
  - – (Two Hypens) - This is a comment that begins with two hyphens and ends with end of line.
  - {} - Braces (Braces enclose the comment. The { precedes the comment, and the } follows it. Braces can delimit single- or multiple-line comments, but comments cannot be nested)
  - `/**/: C style comments (Does not allow nested comments). Please note /*! <comment that begin with slash followed by asterisk and exclamation mark is not a comment > */`
  - MySQL Server supports some variants of C-style comments. These enable you to write code that includes MySQL extensions, but is still portable, by using comments of the following form: `/*! MySQL-specific code */`
  - . # : Mysql comments : This is a comment that begins with # character and ends with end of the line
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server. For example; – (Two Hypens), and **/* */** (Allows nested comments)
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.
- **Check all Comments**—Check the entire request for injected SQL without skipping anything. This is the default setting.

> **Tip**
>
> In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Check Request headers—Enable this option if, in addition to examining the input in the form fields, you want to examine the request headers for HTML SQL Injection attacks. If you use the GUI, you can enable this parameter in the **Advanced Settings** -> **Profile Settings** pane of the App Firewall profile.

> **Note:**
>
> If you enable the Check Request header flag, you might have to configure relaxation rule for the **User-Agent** header. Presence of the SQL keyword **like** and SQL special character semi-colon (**;**) might trigger false positive and block requests that contain this header.
> **Warning:**

> If you enable both request header checking and transformation, any SQL special characters found in headers are also transformed. The Accept, Accept-Charset, Accept-Encoding, Accept-Language, Expect, and User-Agent headers normally contain semicolons (;). Enabling both Request header checking and transformation simultaneously might cause errors.

Check Request Query Non HTML—Enable this option if you want to examine the request query portion for SQL Injection attacks irrespective of the content-type. If you use the GUI, you can enable this parameter in the Advanced Settings -> Profile Settings pane of the App Firewall profile.

SQL fine grained relaxations

The App Firewall gives you an option to exempt a specific form field, header, or Cookie from SQL Injection inspection check. You can completely bypass the inspection for one or more of these fields by configuring relaxation rules for the SQL Injection check.

The App Firewall allows you to implement tighter security by fine tuning the relaxation rules. An application might require the flexibility to allow specific patterns, but configuring a relaxation rule to bypass the security inspection might make the application vulnerable to attacks, because the target field is exempted from inspection for any SQL attack patterns. SQL fine grained relaxation provides the option to allow specific patterns and block the rest. For example, the App Firewall currently has a default set of more than 100 SQL keywords. Because hackers can use these keywords in SQL Injection attacks, the App Firewall flags them as potential threats. You can relax one or more keyword(s) that are considered safe for the specific location. The rest of the potentially dangerous SQL keywords are still checked for the target location and continue to trigger the security check violations. You now have much tighter control.

The commands used in relaxations have optional parameters for **Value Type** and **Value Expression**. You can specify whether the value expression is a regular expression or a literal string. The value type can be left blank or you have an option to select **Keyword** or **SpecialString** or **WildChar**.

> **Warning:**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the HTML SQL Injection check would otherwise have blocked.

Points to Consider:

- Value expression is an optional argument. A field name might not have any value expression.
- A field name can be bound to multiple value expressions.

- Value expressions should be assigned a value type. The SQL value type can be: 1) Keyword, 2) SpecialString, or 3) WildChar.
- You can have multiple relaxation rules per field name/URL combination.

Using the command line to configure the SQL injection check

To configure SQL Injection actions and other parameters by using the command line:

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the SQL Injection protections. You can enable the block, learn, log, stats action(s) and specify whether you want to transform the special characters used in SQL Injection attack strings to disable the attack. Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads, and indicate whether you want the App Firewall to also inspect the request Headers for SQL Injection violations. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

- set appfw profile "Parameter descriptions provided at bottom of the page."
- `<name> -SQLInjectionAction (([block] [learn] [log] [stats])| [none])`
- set appfw profile "Parameter descriptions provided at bottom of the page."
- `<name> -SQLInjectionTransformSpecialChars (**ON** | OFF)`
- set appfw profile "Parameter descriptions provided at bottom of the page."
- `<name> -**SQLInjectionCheckSQLWildChars** (**ON** |**OFF**)`
- set appfw profile "Parameter descriptions provided at bottom of the page."
- `**<name> -**SQLInjectionType** ([**SQLKeyword**] | [**SQLSplChar**] | [**SQLSplCharANDKeyword**] | [**SQLSplCharORKeyword**])`
- **set appfw profile** "Parameter descriptions provided at bottom of the page."
- `<name> -**SQLInjectionParseComments** ([**checkall**] | [**ansi|nested **] | [**ansinested**])`
- **set appfw profile "Parameter descriptions provided at bottom of the page."
- `<name> -CheckRequestHeaders (ON | OFF)`
- `<name> -CheckRequestQueryNonHtml (ON | OFF)`

To configure a SQL Injection relaxation rule by using the command line

Use the bind or unbind command to add or delete binding, as follows:

- `bind appfw profile <name> -SQLInjection <String> [isRegex(REGEX| NOTREGE)] <formActionURL> [-location <location>] [-valueType (Keywor |SpecialString|Wildchar)[<valueExpression>][-isValueRegex (REGEX | NOTREGEX)]]`

- `unbind appfw profile <name> -SQLInjection <String> <formActionURL> [-location <location>] [-valueTyp (Keyword|SpecialString|Wildchar)[<`

```
    valueExpression>]]
```

> **Note:**
>
> You can find the list of SQL keywords from default signature file contents by viewing the view signature object, which has list of sql key words and sql special characters.

Using the GUI to configure the SQL injection security check

In the GUI, you can configure the SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the SQL Injection check by using the GUI

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

1. If you just want to enable or disable Block, Log, Stats, and Learn actions for HTML SQL Injection, you can select or clear check boxes in the table, click **OK**, and then click **Save** and **Close** to close the **Security Check** pane.
2. If you want to configure additional options for this security check, double click HTML SQL Injection, or select the row and click Action Settings, to display the following options:

Transform SQL Special character—Transform any SQL Special characters in the request.

Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

Check Request Containing—Type of SQL injection (SQLKeyword, SQLSplChar, SQLSplCharANDKeyword, or SQLSplCharORKeyword) to check.

SQL Comments Handling—Type of comments (Check All Comments, ANSI, Nested, or ANSI/Nested) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

1. To enable or disable the **Check request Header** setting, in the Advanced Settings pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK.** You can either use the X icon at the top right hand side of the Profile Settings pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **App Firewall > Profile**.

2. To enable or disable the **Check request Query Non HTML** setting, in the **Advanced Settings** pane, click **Profile Settings**. In **Common Settings**, Select or clear the **Check Request Headers** check box. Click **OK**. You can either use the X icon at the top right hand side of the **Profile Settings** pane to close this section or, if you have finished configuring this profile, you can click **Done** to return to the **App Firewall > Profile**.

To configure an SQL Injection relaxation rule by using the GUI

- Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
- In the **Advanced Settings** pane, click **Relaxation Rules**.
- In the Relaxation Rules table, double-click the **HTML SQL Injection** entry, or select it and click **Edit**.
- In the **HTML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

> **Note**
>
> When you add a new rule, the **Value Expression** field is not displayed unless you select **Keyword** or **SpecialString** or **WildChar** option in the **Value Type** Field.

To manage SQL Injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **HTML SQL Injection** row and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the SQL Injection patterns by using the GUI

You can use the GUI to view or customize the SQL patterns.

The default SQL patterns are specified in **Application Firewall**> **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default SQL patterns specified in the Default Signatures object will be used by the profile for the SQL Injection security check processing. The rules and patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these patterns, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the SQL patterns in the new User-defined signature object and use this signature object in your profile that is processing the traffic for which you want to use these customized SQL patterns.

For more information, see Signatures

1. **To view default SQL patterns:**

```
1.  Navigate to **Application Firewall > Signatures**, select ***Default Signatures**, and click **Edit**.

```

```
Then click **Manage SQL/XSS patterns**.
```

The **Manage SQL/XSS paths** table shows following four rows pertaining to SQL Injection:

```
     -  Injection (not_alphanum, SQL)/ Keyword
     -  Injection (not_alphanum, SQL)/ specialstring
     -  Injection (not_alphanum, SQL)/ transformrules/transform
     -  Injection (not_alphanum, SQL)/ wildchar

1.  Select a row and click **Manage Elements** to display the corresponding SQL patterns (keywords, special strings, transformation rules or the wildcard characters) used by the App Firewall SQL injection check.
```

To customize SQL patterns: You can edit the User-Defined signature object to customize the SQL key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the SQL special strings.

1. **Navigate to Application Firewall > Signatures**, highlight the target User-defined signature, and click **Edit**. Click **Manage SQL/XSS patterns** to display the **Manage SQL/XSS paths** table.
2. Select the target SQL Injection row.

   i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding SQL element.

   ii. Click **Remove** to remove the selected row.

> **Warning:**
>
> You must be very careful before you remove or modify any default SQL element, or delete the SQL path to remove the entire row. The signature rules as well as the SQL Inject security check rely on these elements for detecting SQL Injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to SQL attacks if the required pattern is removed during editing.

Using the Learn feature with the SQL injection check

When the learn action is enabled, the App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as an SQL Injection relaxation rule.

SQL Injection Learning enhancement—An App Firewall learning enhancement was introduced in release 11.0 of the NetScaler software. To deploy fine grained SQL Injection relaxation, the App Firewall offers fine grained SQL Injection learning. The learning engine makes recommendations regarding the observed Value Type (keyword, SpecialString, Wildchar) and the corresponding Value expression observed in the input fields. In addition to checking the blocked requests to determine whether the current rule is too restrictive and needs to be relaxed, you can review the rules generated by the learning engine to determine which value type and value expressions are triggering violations and need to be addressed in relaxation rules.

> **Important:**
>
> The App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields from SQL Injection inspection.
> **Note** To bypass SQL check in User-Agent header, use the following relaxation rule:
>
> bind appfw profile your_profile_name -SQLInjection User-Agent " .*" -location HEADER

To view or use learned data by using the command line interface

At the command prompt, type one of the following commands:

- `show appfw learningdata <profilename> SQLInjection`
- `rm appfw learningdata <profilename> -SQLInjection <string> <formActionURL> [<location>] [<valueType> <valueExpression>]`
- `export appfw learningdata <profilename> SQLInjection`

To view or use learned data by using the GUI

1. Navigate to **Application Firewall > Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the **HTML SQL Injection** entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the **HTML SQL Injection** entry in the Learned Rules table and clicking **Visualizer** to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the SQL injection check

When the log action is enabled, the HTML SQL Injection security check violations are logged in the audit log as **APPFW_SQL** violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the `/var/log/` folder to access the log messages pertaining to the SQL Injection violations:

```
> Shell
```

```
## tail -f /var/log/ns.log | grep APPFW_SQL
```

Example of a HTML SQL Injection log message when the request is transformed.

```
Jun 26 21:08:41 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|
APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=54001 method
=GET request=http://aaron.stratum8.net/FFC/login.php?login_name\=%27+
or&passwd\=and+%3B&drinking_pref\=on&text_area\=select+*+from+%5C+%3B&
loginButton\=ClickToLogin&as_sfid\=AAAAAAXjnGN5gLH-hvhTOpIySEIqES7BjFRs5Mq0fwPp
-3ZHDi5yWlRWByj0cVbMyy-Ens2vaaiULKOcUri4OD4kbXWwSY5s7I3QkDsrvIgCYMC9BMvBwY2wbNcSqCwk52
%3D&as_fid\=feeec8758b41740eedeeb6b35b85dfd3d5def30c msg= Special characters
 seen in fields cn1=74 cn2=762 cs1=pr_ffc cs2=PPE1 cs3=9ztIlf9p1H7p6Xtzn6NMygTv
/QM0002 cs4=ALERT cs5=2015 act=transformed
```

Example of a HTML SQL Injection log message when the post request is blocked

```
Jun 26 21:30:34 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|
APPFW|APPFW_SQL|6|src=10.217.253.62 geolocation=Unknown spt=9459 method=
POST request=http://aaron.stratum8.net/FFC/login_post.php msg=SQL Keyword
check failed for field text_area\="(')" cn1=78 cn2=834 cs1=pr_ffc cs2=PPE1
cs3=eVJMMPtZ2XgylGrHjkx3rZLfBCI0002 cs4=ALERT cs5=2015 act=blocked
```

> **Note:**
>
> As part of the streaming changes in 10.5.e build (enhancement builds) as well as 11.0 build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input could be separated by a large number of bytes. We now keep track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message now includes the SQL keyword, or the SQL special character, or both the SQL keyword and the SQL special character, as determined by the configured setting. The rest of the input is no longer included in the log message, as shown in

> the following example:
>
> **Example:**
>
> In 10.5, when the App Firewall detects the SQL violation, the entire input string might be included in the log message, as shown below:
>
> ```
> SQL Keyword check failed for field text=\"select a name from testbed1
> ;(;)\".*<blocked>
> ```
>
> In enhancement builds of 10.5.e that support request side streaming as well as 11.0 build onwards, we log only the field name, keyword and special character (if applicable) in the log message, as shown below:
>
> ```
> SQL Keyword check failed for field **text="select(;)"<blocked>
> ```
>
> This change is applicable to requests that contain application/x-www-form-urlencoded, or multipart/form-data, or text/x-gwt-rpc content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected by this change.

To access the log messages by using the GUI

The NetScaler GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **HTML SQL Injection** row and click **Logs**. When you access the logs directly from the HTML SQL Injection check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **HTML SQL Injection** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_SQL** check box and click the **Apply** button, only log messages pertaining to the **SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** etc. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click to Deploy functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy HTML SQL Injection relaxation rules based on the log messages for the App Firewall security check violations. The log messages must be in CEF log format for this operation. Click to deploy functionality is available only for log messages that are generated by the block (or not block) action. You cannot deploy a relaxation rule for a log message about the transform operation.

To deploy a relaxation rule from the Syslog Viewer, select the log message. A check box appears in the upper right corner of the Syslog Viewer box of the selected row. Select the check box, and then select an option from the Action list to deploy the relaxation rule. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options.

The SQL Injection rules that are deployed by using the Click to Deploy option do not include the fine grain relaxation recommendations.

To use Click to Deploy functionality in the GUI:

1. In the Syslog Viewer, select **Application Firewall** in the **Module** options.
2. Select the **APP_SQL** as the **Event Type** to filter corresponding log messages.
3. Select the check box to identify the rule to deploy.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

Statistics for the SQL Injection violations

When the stats action is enabled, the counter for the SQL Injection check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 SQL Injection violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display SQL Injection check statistics by using the command line:

At the command prompt, type:

> **sh appfw stats**

To display stats for a specific profile, use the following command:

```
>  stat appfw profile <profile name>
```

To display HTML SQL Injection statistics by using the GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.

3. Use the scroll bar to view the statistics about HTML SQL Injection violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the SQL Injection check:

- **Built-in Support for SQL Injection Protection**—The NetScaler App App Firewall protects against SQL Injection by monitoring a combination of SQL keywords and special characters in the form parameters. All SQL keywords, special characters, wildcard characters, and default transformation rules are specified in the /netscaler/default_custom_settings.xml file.
- **Customization**—You can change the default key words, special characters, wildcard characters and transformation rules to customize the SQL security check inspection for the specific needs of your application. Make a copy of the default signature object, modify existing entries, or add new ones. Bind this signature object to your profile to make use of the customized configuration.
- **Hybrid Security Model**—Both signatures and deep security protections use the SQL/XSS patterns specified in the signature object that is bound to the profile. If no signature object is bound to the profile, the SQL/XSS patterns present in the default signature object are used.
- **Transform**—Note the following about the transform operation:
  - The transform operation works independently of the other SQL Injection action settings. If transform is enabled and block, log, stats, and learn are all disabled, SQL special characters will be transformed.
  - When SQL Transformation is enabled, user requests are sent to the backend servers after the SQL special characters are transformed in non-block mode. If the block action is enabled, it takes precedence over the transform action. If the injection type is specified as SQL special character and block is enabled, the request is blocked despite the transform action.
- **Fine Grained Relaxation and Learning**—Fine tune the relaxation rule to relax a subset of SQL elements from security check inspection but detect the rest. The learning engine recommends a specific value type and value expressions based on the observed data.
- **Click to Deploy**—Select one, or multiple SQL violation log messages in the syslog viewer and deploy them as relaxation rules.

Buffer overflow check

September 21, 2020

The Buffer Overflow check detects attempts to cause a buffer overflow on the web server. If the App

Firewall detects that the URL, cookies, or header are longer than the specified maximum length in a request, it blocks that request because it might be an attempt to cause a buffer overflow.

The Buffer Overflow check prevents attacks against insecure operating-system or web-server software that can crash or behave unpredictably when it receives a data string that is larger than it can handle. Proper programming techniques prevent buffer overflows by checking incoming data and either rejecting or truncating overlong strings. Many programs, however, do not check all incoming data and are therefore vulnerable to buffer overflows. This issue especially affects older versions of web-server software and operating systems, many of which are still in use.

The Buffer Overflow security check allows you to configure the **Block**, **Log**, and **Stats** actions. In addition, you can also configure the following parameters:

- **Maximum URL Length.** The maximum length the App Firewall allows in a requested URL. Requests with longer URLs are blocked. **Possible Values**: 0-65535. **Default**: 1024
- **Maximum Cookie Length.** The maximum length the App Firewall allows for all cookies in a request. Requests with longer cookies trigger the violations. **Possible Values**: 0-65535. **Default**: 4096
- **Maximum Header Length.** The maximum length the App Firewall allows for HTTP headers. Requests with longer headers are blocked. **Possible Values**: 0-65535. **Default**: 4096

Using the command line to configure the Buffer Overflow security check

To configure Buffer Overflow security check actions and other parameters by using the command line

If you use the command-line interface, you can add the following Buffer Overflow Check arguments to the set appfw profile `<profileName>` command:

- `-bufferOverflowAction [[ block] [ log] [stats]] | [none]`
- `-bufferOverflowMaxURLLength <positiveInteger>`
- `-bufferOverflowMaxCookieLength <positiveInteger>`
- `-bufferOverflowMaxHeaderLength <positiveInteger>`

Using the GUI to Configure the Buffer Overflow Security Check

In the GUI, you can configure the Buffer Overflow security check in the pane for the profile associated with your application.

To configure or modify the Buffer Overflow security check by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you just want to enable or disable **Block**, **Log**, and **Stats** actions for **Buffer Overflow**, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

b. If you want to configure additional options for this security check, double click **Buffer Overflow**, or select the row and click **Action Settings**, to display the following options:

Maximum URL Length.

Maximum Cookie Length.

Maximum Header Length.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

Using the Log Feature with the Buffer Overflow Security Check

When the log action is enabled, the Buffer Overflow security check violations are logged in the audit log as **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and **APPFW_BUFFEROVERFLOW_HDR** violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

If you use the GUI to review the logs, you can use the click-to-deploy feature to apply relaxations indicated by the logs.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Buffer overflow violations:

```
> **Shell**
> **tail -f /var/log/ns.log | grep APPFW_BUFFEROVERFLOW**
<!--NeedCopy-->
```

Example of a CEF log message showing bufferOverflowMaxCookieLength violation in non-block mode

```
Oct 22 17:35:20 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_BUFFEROVERFLOW_COOKIE|6|src=10.217.253.62 geolocation=Unknown spt=41198 method=GET request=http://aaron.stratum8.net/FFC/sc11.html msg=Cookie header length(43) is greater than maximum allowed(16). cn1=119 cn2=465 cs1=owa_profile cs2=PPE1 cs3=wvOOOb+cJ2ZRbstZpyeNXIqLj7Y0001 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

Example of a CEF log message showing bufferOverflowMaxURLLength violation in non-block mode

```
Oct 22 18:39:56 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_BUFFEROVERFLOW_URL |6|src=10.217.253.62 geolocation=Unknown spt=19171 method=GET request=http://aaron.stratum8.net/FFC/sc11.html **msg=URL length(39) is greater than maximum allowed(20). cn1=707 cn2=402 cs1=owa_profile cs2=PPE0 cs3=kW49GcKbnwKByByi3+jeNzfgWa80000 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

Example of a Native Format Log message showing bufferOverflowMaxHeaderLength violation in block mode

```
Oct 22 18:44:00 <local0.info> 10.217.31.98 10/22/2015:18:44:00 GMT ns 0-PPE-2 : default APPFW APPFW_BUFFEROVERFLOW_HDR 155 0 : 10.217.253.62 374-PPE2 khhBEeY4DB8V2D3H2sMLkXmfWnA0002 owa_profile Header(User-Agent) length(82) is greater than maximum allowed(10)** : http://aaron.stratum8.net/ <blocked>
<!--NeedCopy-->
```

To access the log messages by using the GUI

The NetScaler GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **Buffer Overflow** row and click **Logs**. When you access the logs directly from the Buffer Overflow Security Check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **Buffer Overflow** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers three options, **APPFW_BUFFEROVERFLOW_URL**, **APPFW_BUFFEROVERFLOW_COOKIE**, and **APPFW_BUFFEROVERFLOW_HDR**, to view all the log messages pertaining to buffer overflow security check. You can select one or more options to further refine your selection. For example, if you select the **APPFW_BUFFEROVERFLOW_COOKIE** check box and click the **Apply** button, only log messages pertaining to the Buffer Overflow security check violations for the Cookie header appear in the Syslog Viewer. If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and **Client IP**, appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Click-to-Deploy: The GUI provides click-to-deploy functionality, which is currently supported only for the buffer overflow log messages pertaining to the **URL Length** violations. You can use the Syslog Viewer to not only view the triggered violations, but also execute informed decisions based on the observed lengths of the blocked messages. If the current value is too restrictive and is triggering false positives, you can select a message and deploy it to replace the current value with the URL length value seen in the message. The log messages must be in CEF log format for this operation. If the relaxation can be deployed for a log message, a check box appears at the right edge of the Syslog Viewer box in the row. Select the check box, and then select an option from the **Action** list to deploy the relaxation. **Edit & Deploy**, **Deploy**, and **Deploy All** are available as **Action** options. You can use the **APPFW_BUFFEROVERFLOW_URL** filter to isolate all the log messages pertaining to the configured URL length violations.

If you select an individual log message, all three action options **Edit & Deploy**, **Deploy**, and **Deploy All** are available. If you select **Edit & Deploy**, the **Buffer Overflow settings** dialogue is displayed. The new URL length that was observed in the request is inserted into the Maximum URL length input field. If you click **Close** without any edits, the current configured values remain unchanged. If you click the **OK** button, the new value of the Maximum URL length replaces the previous value.

> **Note**
>
> The **block**, **log** and **stats** action check boxes are unchecked in the displayed **Buffer Overflow settings** dialogue, and need to be reconfigured if you select the **Edit & Deploy** option. Make sure to enable these check boxes before clicking **OK**, otherwise the new URL length will get configured but the actions will be set to **none**.

If you select the check boxes for multiple log messages, you can use the **Deploy** or **Deploy All** option. If the deployed log messages have different URL lengths, the configured value gets replaced by the highest URL Length value observed in the selected messages. Deploying the rule results only in changing the **bufferOverflowMaxURLLength** value. Configured actions are retained and remain unchanged.

To use Click-to-Deploy functionality in the GUI

1. In the Syslog Viewer, select **APPFW** in the **Module** options.
2. Enable the **APPFW_BUFFEROVERFLOW_URL** check box as the **Event Type** to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the **Action** drop-down list of options to deploy the relaxation.

5. Navigate to **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks** to access the **Buffer Overflow** settings pane to verify that the **Maximum URL Length** value is updated.

Statistics for the Buffer Overflow violations

When the stats action is enabled, the counter for the Buffer Overflow Security Check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three Buffer Overflow violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display Buffer Overflow Security Check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> stat appfw profile <profile name>
```

To display Buffer Overflow statistics by using the GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about Buffer Overflow violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

- The buffer overflow security check allows you to configure limits to enforce the maximum length of allowed URLs, Cookies and Headers.
- **Block**, **Log** and **Stats** actions enable you to monitor the traffic and configure optimal protection for your application.
- Syslog viewer enables you to filter and view all the log messages pertaining to buffer overflow violations.
- **Click-to-Deploy** functionality is supported for the **bufferOverflowMaxURLLength** violations. You can select and deploy an individual rule, or you can select multiple log messages to tweak and relax the current configured value of the maximum allowed length of the URL. The highest

value of the URL from the selected group is set as the new value, to allow all these requests that are currently flagged as violations.

- The App Firewall now evaluates individual cookies when inspecting the incoming request. If length of any one cookie received in the Cookie header exceeds the configured **BufferOverflowMaxCookieLength**, the Buffer Overflow violation is triggered.

> **Important**
>
> In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), App Firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. ** In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".
>
> **Note** that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.**

Cookie consistency check

September 21, 2020

The Cookie Consistency check examines cookies returned by users, to verify that they match the cookies that your web site set for that user. If a modified cookie is found, it is stripped from the request before the request is forwarded to the web server. You can also configure the Cookie Consistency check to transform all of the server cookies that it processes, by encrypting the cookies, proxying the cookies, or adding flags to the cookies. This check applies to requests and responses.

An attacker would normally modify a cookie to gain access to sensitive private information by posing as a previously authenticated user, or to cause a buffer overflow. The Buffer Overflow check protects against attempts to cause a buffer overflow by using a very long cookie. The Cookie Consistency check focuses on the first scenario.

If you use the wizard or the GUI, in the
Modify Cookie Consistency Check dialog box, on the
General tab you can enable or disable the following actions:

- Block
- Log
- Learn
- Statistics
- Transform. If enabled, the Transform action modifies all cookies as specified in the following settings:
  - **Encrypt Server Cookies.** Encrypt cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list, before forwarding the response to the client. Encrypted cookies are decrypted when the client sends a subsequent request, and the decrypted cookies are reinserted into the request before it is forwarded to the protected web server. Specify one of the following types of encryption:
    - **None.** Do not encrypt or decrypt cookies. The default.
    - **Decrypt only.** Decrypt encrypted cookies only. Do not encrypt cookies.
    - **Encrypt session only.** Encrypt session cookies only. Do not encrypt persistent cookies. Decrypt any encrypted cookies.
    - **Encrypt all.** Encrypt both session and persistent cookies. Decrypt any encrypted cookies.
      Note: When encrypting cookies, the App Firewall adds the
      HttpOnly flag to the cookie. This flag prevents scripts from accessing and parsing the cookie. The flag therefore prevents a script-based virus or trojan from accessing a decrypted cookie and using that information to breach security. This is done regardless of the
      Flags to Add in Cookies parameter settings, which are handled independently of the Encrypt Server Cookies parameter settings.
- **Proxy Server Cookies.** Proxy all non-persistent (session) cookies set by your web server, except for any listed in the Cookie Consistency check relaxation list. Cookies are proxied by using the existing App Firewall session cookie. The App Firewall strips session cookies set by the protected web server and saves them locally before forwarding the response to the client. When the client sends a subsequent request, the App Firewall reinserts the session cookies into the request before forwarding it to the protected web server. Specify one of the following settings:
  - **None.** Do not proxy cookies. The default.
  - **Session only.** Proxy session cookies only. Do not proxy persistent cookies
    Note: If you disable cookie proxying after having enabled it (set this value to
    None after it was set to
    Session only), cookie proxying is maintained for sessions that were established before you disabled it. You can therefore safely disable this feature while the App Firewall is processing user sessions.
- **Flags to Add in Cookies**. Add flags to cookies during transformation. Specify one of the following settings:

  - **None.** Do not add flags to cookies. The default.
  - **HTTP only.** Add the HttpOnly flag to all cookies. Browsers that support the HttpOnly flag do not allow scripts to access cookies that have this flag set.
  - **Secure**. Add the Secure flag to cookies that are to be sent only over an SSL connection. Browsers that support the Secure flag do not send the flagged cookies over an insecure connection.
  - **All.** Add the HttpOnly flag to all cookies, and the Secure flag to cookies that are to be sent only over an SSL connection.

If you use the command-line interface, you can enter the following commands to configure the Cookie Consistency Check:

- `set appfw profile <name> -cookieConsistencyAction [**block**] [**learn**] [**log**] [**stats**] [**none**]`
- `set appfw profile <name> -cookieTransforms ([**ON**] | [**OFF**])`
- `set appfw profile <name> -cookieEncryption ([**none**] | [**decryptOnly**] | [**encryptSession**] | [**encryptAll**])`
- `set appfw profile <name> -cookieProxying ([**none**] | [**sessionOnly**])`
- `set appfw profile <name> -addCookieFlags ([**none**] | [**httpOnly**] | [**secure**] | [**all**])`

To specify relaxations for the Cookie Consistency check, you must use the GUI. On the Checks tab of the Modify Cookie Consistency Check dialog box, click Add to open the Add Cookie Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Cookie Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Cookie Consistency check relaxations:

- **Logon Fields.** The following expression exempts all cookie names beginning with the string logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^logon_[0-9A-Za-z]{
2,15 }
$
<!--NeedCopy-->
```

- **Logon Fields (special characters).** The following expression exempts all cookie names beginning with the string türkçe-logon_ followed by a string of letters or numbers that is at least two characters long and no more than fifteen characters long:

```
^txC3xBCrkxC3xA7e-logon_[0-9A-Za-z]{
```

```
2,15 }
$
<!--NeedCopy-->
```

- **Arbitrary strings.** Allow cookies that contain the string sc-item_, followed by the ID of an item that the user has added to his shopping cart ([0-9A-Za-z]+), a second underscore (_), and finally the number of these items he wants ([1-9][0-9]?), to be user-modifiable:

```
^sc-item_[0-9A-Za-z]+_[1-9][0-9]?$
<!--NeedCopy-->
```

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (
.*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

> **Important**
>
> In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), App Firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".
>
> **Note** that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.**
>
> **Note**
>
> **Sessionless Cookie Consistency:** The cookie consistency behavior has changed in release 11.0. In earlier releases, the cookie consistency check invokes sessionization. The cookies are stored in the session and signed. A "wlt_" suffix is appended to transient cookies and a "wlf_" suffix is appended to the persistent cookies before they are forwarded to the client. Even if the client

> does not return these signed wlf/wlt cookies, the App Firewall uses the cookies stored in the session to perform the cookie consistency check.
>
> In release 11.0, the cookie consistency check is sessionless. The App Firewall now adds a cookie that is a hash of all the cookies tracked by the App Firewall. If this hash cookie or any other tracked cookie is missing or tampered with, the App Firewall strips the cookies before forwarding the request to the back end server and triggers a cookie-consistency violation. The server treats the request as a new request and sends new Set-Cookie header(s).
> The Cookie Consistency check in Citrix ADC version 13.0, 12.1, and NetScaler 12.0 and 11.1 does not have sessionless option.

App Firewall support for Google web toolkit

September 21, 2020

Note: This feature is available in NetScaler release 10.5.e.

Web servers following Google Web Toolkit (GWT) Remote Procedure Call (RPC) mechanisms can be secured by the NetScaler App App Firewall without a need for any specific configuration to enable the GWT support.

What is GWT

The GWT is used for building and optimizing complex high-performance web applications by people who do not have expertise in XMLHttpRequest, and JavaScript. This open source, free development toolkit is used extensively for developing small and large scale applications and is quite frequently used for displaying browser based data such as search results for flights, hotels, and so on. The GWT provides a core set of Java APIs and widgets for writing optimized JavaScript scripts that can run on most browsers and mobile devices. The GWT RPC framework makes it easy for the client and server components of the web application to exchange Java objects over HTTP. GWT RPC services are not the same as web services based on SOAP or REST. They are simply a lightweight method for transferring data between the server and the GWT application on the client. GWT handles serialization of the Java objects exchanging the arguments in the method calls and the return value.

For popular websites that use GWT, see

https://www.quora.com/What-web-applications-use-Google-Web-Toolkit-%28GWT%29

How a GWT request works

The GWT RPC request is pipe delimited and has variable number of arguments. It is carried as a payload of HTTP POST and has the following values:

1. Content-type = text/x-gwt-rpc. Charset can be any value.
2. Method = POST.

Both GET and POST HTTP requests are considered valid GWT requests if content-type is "text/x-gwt-rpc".
Query strings are now supported as part of GWT requests.

The following example shows a valid payload for a GWT request:

```
5|0|8|http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.TestService|testMethod|java.lang.String|java.lang.Integer| myInput1|java.lang.Integer/3438268394|1|2|3|4|2|5|6|7|8|1|
<!--NeedCopy-->
```

The request can be divided into three parts:

**a) `Header: 5|0|8|**`

The first 3 digits "5	0	8	" in the above request, represent "version, subversion, and size of table", respectively. These must be positive integers.

b) String Table:

`http://localhost:8080/test/|16878339F02B83818D264AE430C20468| com.test.client.TestService|testMethod|java.lang.String|java.lang.Integer|myInput1|java.lang.Integer/3438268394|`

The members of the above pipe delimited string table contain the user-provided inputs. These inputs are parsed for the App Firewall checks and are identified as follows:

- 1st : `http://localhost:8080/test/`

  This is the Request URL.

- 2nd : 16878339F02B83818D264AE430C20468

  Unique HEX identifier. A request is considered malformed if this string has non-hex characters.

- rd : com.test.client.TestService

  Service Class name

- 4th : testMethod

  Service method name

5th onwards: java.lang.String	java.lang.Integer	myInput1	java.lang.Integer/3438268394

- Data-types and data. Non-primitive data-types are specified as

  `<container>.<sub-cntnr>.name/<integer><identifier>`

**c) `Payload: 1|2|3|4|2|5|6|7|8|1|**`

The payload consists of references to the elements in the string table. These integer values cannot be larger than the number of elements in the string table.

App Firewall protection for GWT applications

The App Firewall understands and interprets GWT RPC requests, inspects the payload for security check violations, and takes specified actions.

The App Firewall headers and cookies checks for GWT requests are similar to those for other request formats. After appropriate URL decoding and charset conversion, all the parameters in the string table are inspected. The GWT request body does not contain field names, just the field values. The input values can be validated against the specified format by using the App Firewall Field Format check, which can also be used to control the length of the input. The **Cross-site Scripting** and **SQL Injection** attacks in the inputs can be easily detected and thwarted by the App Firewall.

Learning and relaxation rules: Learning and deployment of relaxation rules are supported for GWT requests. App Firewall rules are in the form of `<actionURL>` `<fieldName>` mapping. The GWT request format does not have the field names and thus requires special handling. The App Firewall inserts dummy field names in the learned rules that can be deployed as relaxation rules. The -isRegex flag works as it does for non-GWT rules.

- Action URL:

  Multiple services responding to an RPC can be configured on the same web server. The HTTP request has the URL of the web server, not of the actual service handling the RPC. Therefore, relaxation is not applied on the basis of the HTTP request URL, because that would relax all the services on that URL for the target field. For GWT requests, the App Firewall uses the URL of the actual service found in the GWT payload, in the fourth field in the string table.

- Field name:

  Since the GWT request body contains only field values, the App Firewall inserts dummy field names such as 1, 2, and so on when recommending learned rules.

Example of a GWT learned rule

```
POST /abcd/def/gh HTTP/1.1
Content-type: text/x-gwt-rpc
Host: 10.217.222.75
Content-length: 157

5|0|8|http://localhost:8080/acdtest/|16878339
    F02Baf83818D264AE430C20468|
com.test.client.TestService|testMethod|java.lang.String%3b|java.
    lang.Integer|onblur|

  The learn data will be as follows:
> sh learningdata pr1 crossSiteScripting
Profile: pr1    SecurityCheck: crossSiteScripting
1)  Url:    http://localhost:8080/acdtest/  >> From GWT Payload.
    Field:  10
    Hits:   1
 Done
<!--NeedCopy-->
```

Example of a GWT relaxation rule

bind appfw profile pr1 -crossSiteScripting 1 abcd -isregex NOTREGEX

Log Messages: The App Firewall generates log messages for the security check violations that are detected in the GWT requests. A log message generated by a malformed GWT request contains the string "GWT" for easy identification.

Example of a Log message for malformed GWT request:

```
Dec 5 21:48:02 <local0.notice> 10.217.31.247 12/05/2014:21:48:02 GMT ns
0-PPE-0 : APPFW Message 696 0 : "GWT RPC request with malformed payload. <
blocked>"
```

Difference in processing of GWT vs non-GWT requests:

The same payload can trigger different App Firewall security check violations for different Content-types. Consider the following example:

```
5|0|8|http://localhost:8080/acdtest/|16878339F02Baf83818D264AE430C20468|com
.test.client.TestService|testMethod|java.lang.String%3b|java.lang.Integer|
select|
```

Content-type: application/x-www-form-urlencoded:

A request sent with this content type results in a SQL violation if the SQL Injection Type is configured to use any of the four available options: SQLSplCharANDKeyword, SQLSplCharORKeyword, SQLKeyword, or SQLSplChar. The App Firewall considers '&' to be the field separator and '=' to be the name-value separator when processing the above payload. Since neither of these characters appears anywhere in the post body, the entire content is treated as a single field name. The field name in this request contains both an SQL special character (;) and an SQL Keyword (select). Therefore violations are caught for all four SQL Injection type options.

Content-type: text/x-gwt-rpc:

A request sent with this content type triggers an SQL violation only if the SQL injection type is set to one of the following three options:SQLSplCharORKeyword,SQLKeyword, or SQLSplChar. No violation is triggered if the SQL injection type is set to SQLSplCharANDKeyword, which is the default option. The App Firewall considers the vertical bar '	' to be the field separator for the above payload in the GWT request. Therefore, the post body is divided into various form-field values, and form-field names are added (in accordance with the convention described earlier). Because of this splitting, the SQL special character and SQL keyword become parts of separate form fields.

Form field 8: java.lang.String%3b -> %3b is the (;) char

Form Field 10: select

As a result, when SQL Injection Type is set to **SQLSplChar**, field 8 indicates the SQL violation. For **SQLKeyword**, field 10 indicates the violation. Either of these two fields can indicate a violation if the SQL Inject type is configured with the **SQLSplCharORKeyword** option, which looks for the presence of either a keyword or a special character. No violation is caught for the default **SQLSplCharANDKeyword** option, because there is no single field that has a value that contains both **SQLSplChar** and **SQLKeyword** together.

Tips:

- No special App Firewall configuration is needed to enable GWT support.
- The Content-type must be text/x-gwt-rpc.
- Learning and deploying of the relaxation rules for all the pertinent App Firewall security checks applied to GWT payload works the same as it does for the other supported content-types.
- Only POST requests are considered valid for GWT. All other request methods are blocked if the content-type is text/x-gwt-rpc.
- GWT requests are subject to the configured POST body limit of the profile.
- The sessionless setting for the security checks is not applicable and will be ignored.
- CEF log format is supported for the GWT log messages.

Data leak prevention checks

September 21, 2020

The data-leak-prevention checks filter responses to prevent leaks of sensitive information, such as credit card numbers and social security numbers, to unauthorized recipients.

Credit card check

September 21, 2020

If you have an application that accepts credit cards, or your websites have access to database servers that store credit card numbers, you must use Data Leak Prevention (DLP) measures and configure protection for each type of credit card that you accept.

The NetScaler App App Firewall Credit Card check prevents attackers from exploiting Data Leak Prevention flaws to obtain credit card numbers of your customers. By following simple configuration steps, you can enforce protection of one or more of the following credit cards: 1) Visa, 2) Master Card, 3) Discover, 4) American Express (Amex), 5) JCB, and 6) Diners Club.

The Credit Card security check examines server responses to identify instances of the target credit card numbers, and applies a specified action when such a number is found. The action can be to transform the response by X'ing out all but the last group of digits in the credit card number, or to block the response if it contains more than a specified number of credit card numbers. If you specify both, the block action takes precedence. The Maximum credit cards allowed per page setting determines when the block action is invoked. The default setting, 0 (no credit card numbers allowed on the page), is the safest, but you can allow up to 255. Depending on where the violation is detected in the response and the block action gets triggered, you might get fewer than the maximum allowed number of credit cards in the response.

To avoid false positives, you can apply relaxations to exempt specific numbers from the Credit Card check. For example, a social security number, purchase order number, or Google account number might be similar to a credit card number. You can specify individual numbers or use a regular expression to indicate the string of digits to be bypassed when processing the response URL for credit card inspection.

If you're not sure which credit card numbers to exempt, you can use the learn feature to generate recommendations based on the learned data. To get optimal benefit without compromising performance, you might want to enable this option for a short time to get a representative sample of the rules, and then deploy the relaxations and disable learning.

If you enable the log feature, the Credit Card check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate thwarted attempts to gain access. By default, the doSecureCreditCardLogging parameter is ON, so the credit card number is not included in the log message generated by the safe commerce (Credit Card) violation.

The stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack.

To configure the Credit Card security check for protecting your application, configure the profile that governs inspecting the traffic to and from this application.

> **Note**:
>
> A website that does not access a SQL database usually does not have access to sensitive private information such as credit card numbers.

Using the command line to configure the credit card check

In the command line interface, you can use either the set appfw profile command or the add appfw profile command to activate credit-card checking and specify which actions to perform. You can use the unset appfw profile command to revert back to the default settings. To specify relaxations, use the bind appfw command to bind credit card numbers to the profile.

To configure a credit card check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:

- `set appfw profile <name> -creditCardAction ( ([block][learn] [log][stats])| [none])`
- `set appfw profile <name> -creditCard (VISA | MASTERCARD | DISCOVER | AMEX | JCB | DINERSCLUB)`
- `set appfw profile <name> -creditCardMaxAllowed <integer>`
- `set appfw profile <name> -creditCardXOut ([ON] | [OFF])<name> -doSecureCreditCard ([ON] | [OFF])`
- To configure a Credit Card relaxation rule by using the command line

Use the bind command to bind the credit card number to the profile. To remove a credit card number from a profile, use the unbind command, with the same arguments that you used for the bind command. You can use the show command to display the credit card numbers bound to a profile.

To bind a credit card number a profile

```
`bind appfw profile <profile-name> -creditCardNumber <any number/
   regex> “<url>” `

**Example**: bind appfw profile test_profile -creditCardNumber
   378282246310005 `http://www.example.com/credit\_card\_test.html`

-

To unbind a credit card number from a profile

`unbind appfw profile <profile-name>
-creditCardNumber <credit card number / regex> “<url>`
-

To show the list of credit card numbers bound to a profile.

show appfw profile <profile>
```

Using the GUI to configure the credit card check

In the GUI, you configure the credit card security check in the pane for the profile associated with your application.

To add or modify the Credit Card security check by using the GUI

1. Navigate to **App Firewall** > **Profiles, highlight** the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a) If you just want to enable or disable Block, Log, Stats, and Learn actions for Credit Card, you can select or clear check boxes in the table, click **OK**, and then click **Save** and **Close** to close the **Security Check** pane.
   b) If you want to configure additional options for this security check, double click Credit Card, or select the row and click **Action Settings** to display additional options as follows:
      - Out—Mask any credit card number detected in a response by replacing each digit, except the digits in the final group, with the letter “X.
      - Maximum credit cards allowed per page—Specify the number of credit cards that can be forwarded to the client without triggering a block action.
      - Protected Credit Cards. Select or clear a check box to enable or disable protection for each type of credit card.
      - You can also edit the Block, Log, Stats and Learn actions in the Credit Card Settings pane.

After making any of the above changes, click OK to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click OK to save all the changes you have made in the Security Checks section and then click Save and Close to close the Security Check pane.

3. In the **Advanced Settings** pane, click **Profile settings**. To enable or disable secure logging of credit card Numbers, select or clear the **Secure Credit Card Logging** check box. (By default, it is selected).

   Click **OK** to save the changes.

- To configure a Credit Card relaxation rule by using the GUI

  1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
  2. In the **Advanced Settings** pane, click **Relaxation Rules**. The Relaxation Rules table has a Credit Card entry. You can double click, or select this row and click **Edit** to access the **Credit Card Relaxation Rules** dialogue. You can perform Add, Edit, Delete, Enable, or Disable operations for relaxation rules.

Using the learn feature with the credit card check

When the learn action is enabled, the App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, if you want to exempt a specific string of digits from the Credit Card security check, you can by deploy the learned rule as a relaxation rule.

- To view or use learned data by using the command line interface

  ```
  show appfw learningdata <profilename> creditCardNumber
  ```

  ```
  rm appfw learningdata <profilename> -creditcardNumber <credit card
  number>  “<url>”
  ```

  ```
  export appfw learningdata <profilename> creditCardNumber
  ```

- To view or use learned data by using the GUI

  1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
  2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the Credit Card entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

  You also have the option to show a summarized view of the learned relaxations by selecting the Credit Card entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules.

It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the credit card check

When the log action is enabled, the Credit Card security check violations are logged in the audit log as APPFW_SAFECOMMERCE or APPFW_SAFECOMMERCE_XFORM violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

The default setting for doSecureCreditCardLogging is ON. If you change it to OFF, both credit card number and type are included in the log message.

Depending on the settings configured for the Credit Card checks, the application-firewall generated log messages might include the following information:

- Response was blocked or not blocked.
- Credit card numbers were transformed (X'd out). A separate log message is generated for each transformed credit card number, so multiple log messages might be generated during processing of a single response.
- Response contained the maximum number of potential credit card numbers.
- Credit card numbers and their corresponding types.
- To access the log messages by using the command line

  Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the Credit Card violations:

  - Shell

```
tail -f /var/log/ns.log                                   grep SAFECOMMERCE
```

  -

- To access the log messages by using the GUI
  1. The NetScaler GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have a couple of options for accessing the Syslog Viewer: Navigate to the **target profile** > **Security Checks**. Highlight the Credit Card row and click Logs. When you access the logs directly from the Credit Card security check of the profile, it filters out the

log messages and displays only the logs pertaining to these security check violations.

2. You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.

   The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Credit Card security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the APPFW_SAFECOMMERCE and APPFW_SAFECOMMERCE_XFORM check boxes and click the Apply button, only log messages pertaining to the Credit Card security check violations appear in the Syslog Viewer.

   If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the response is not blocked

```
May 29 01:26:31 <local0.info> 10.217.31.98 05/29/2015:01:26:31 GMT ns0-PPE-0 :
default APPFW APPFW_SAFECOMMERCE 2181 0 :  10.217.253.62 1098-PPE0
4erNfkaHy0IeGP+nv2S9Rsdu77I0000 pr_ffc http://aaron.stratum8.net/FFC/CreditCardMind.html
Maximum number of potential credit card numbers seen <not blocked>
<!--NeedCopy-->
```

Example of a CEF format log message when the response is transformed

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE_XFORM|6|src=10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/CreditCardMind.html
msg=Transformed (xout) potential credit card numbers seen in server response
cn1=66 cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002
cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

Example of a CEF format log message when the response is blocked. The credit card number and type can be seen in the log, because the doSecureCreditCardLogging parameter is disabled.

```
May 28 23:42:48 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_SAFECOMMERCE|6|src
    =10.217.253.62
spt=25314 method=GET request=http://aaron.stratum8.net/FFC/
    CreditCardMind.html
msg=Credit Card number 4505050504030302 of type Visa is seen in
    response cn1=68
cn2=1095 cs1=pr_ffc cs2=PPE2 cs3=xzE7M0g9bovAtG/zLCrLd2zkVl80002 cs4=
    ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

Statistics for the credit card violations

When the stats action is enabled, the corresponding counter for the Credit Card check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled and the Max Allowed credit card setting is 0, the request for a page that contains 20 credit card numbers increments the stats counter by one when the page is blocked as soon as the first credit card number is detected. However, if block is disabled and transform is enabled, processing the same request increments the statistics counter for logs by 20, because each credit card transformation generates a separate log message.

- To display Credit Card statistics by using command line

  At the command prompt, type:

  `sh appfw stats`

  To display stats for a specific profile, use the following command:

  `stat appfw profile <profile name>`

  To display Credit Card statistics by using GUI

  1. Navigate to **System** > **Security** > **App Firewall**.
  2. In the right pane, access the **Statistics** Link.
  3. Use the scroll bar to view the statistics about Credit Card violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Highlights

Note the following points about the Credit Card security check:

- The App Firewall enables you to protect credit card information and detect any attempts to access this sensitive data.
- To use the Credit Card protection check, you must specify at least one type of credit card and an action. The check is then applied to HTML, XML, and Web 2.0 profiles.

You can pipe the output of sh appfw profile command and grep for CreditCard to see all the Credit Card specific configuration. For example, sh appfw profile my_profile grep CreditCard displays the configured settings of various parameters as well as the relaxation rules pertaining to the Credit Card check for the App Firewall profile named my_profile.

-
- You can exclude specific numbers from Credit Card inspection without bypassing the security check inspection for the rest of the credit card numbers.
- Relaxation is available for all App Firewall protected credit card patterns. In the GUI, you can use the visualizer to specify Add, Edit, Delete, Enable, or Disable operations on relaxation rules.
- The App Firewall learning engine can monitor the outgoing traffic to recommend rules based on observed violations. Visualizer support is also available for managing the learned credit card rules in the GUI. You can edit and deploy the learned rules, or skip them after careful inspection.
- The setting for number of allowed credit cards applies to each response. It does not pertain to the cumulative total of credit card numbers observed during the entire user session.
- The number of X'd out digits depends on the length of the credit card numbers. Ten digits are X'd out for credit cards that have 13 through 15 digits. Twelve digits are X'd out for credit cards that have 16 digits. If your application does not require sending the entire credit card number in the response, Citrix recommends that you enable this action to mask the digits in the credit card numbers.
- The X-out operation transforms all the credit cards and works independently of the configured settings for the maximum number of allowed credit cards. For example, if there are 4 credit cards in the response and the creditCardMaxAllowed parameter is set to 10, all 4 credit cards are X'd-out, but they are not blocked. If the credit card numbers are spread out in the document, a partial response with X'd-out numbers might be sent to the client before the response is blocked.
- Do not disable the doSecureCreditCardLogging parameter before due consideration. When this parameter is turned off, the credit card numbers are displayed and are accessible in the log messages. These numbers are not masked in the logs, even if the X-out action is enabled. If you are sending logs to a remote syslog server, and the logs are compromised, the credit card numbers can be exposed.

- When the response page is blocked because of a Credit Card violation, the App Firewall does not redirect to the error page.

Safe object check

September 21, 2020

The Safe Object check provides user-configurable protection for sensitive business information, such as customer numbers, order numbers, and country-specific or region-specific telephone numbers or postal codes. A user-defined regular expression or custom plug-in tells the App Firewall the format of this information and defines the rules to be used to protect it. If a string in a user request matches a safe object definition, the App Firewall either blocks the response, masks the protected information, or removes the protected information from the response before sending it to the user, depending on how you configured that particular safe object rule.

The Safe Object check prevents attackers from exploiting a security flaw in your web server software or on your web site to obtain sensitive private information, such as company credit card numbers or social security numbers. If your web sites do not have access to these types of information, you do not need to configure this check. If you have a shopping cart or other application that can access such information, or your web sites have access to database servers that contain such information, you should configure protection for each type of sensitive private information that you handle and store.

> **Note**:
>
> A web site that does not access an SQL database usually does not have access to sensitive private information.

The Safe Object Check dialog box is unlike that for any other check. Each safe object expression that you create is the equivalent of a separate security check, similar to the Credit Card check, for that type of information. If you use the wizard or the GUI, you add a new expression by clicking Add and configuring the expression in the Add Safe Object dialog box. You modify an existing expression by selecting it, then clicking Open, and then configuring the expression in the Modify Safe Object dialog box.

In the Safe Object dialog box for each safe object expression, you can configure the following:

- **Safe Object Name.** A name for your new safe object. The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 255 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) symbols.
- **Actions.** Enable or disable the
  Block,

Log, and
Statistics actions, and the following actions:

  - **X-Out.** Mask any information that matches the safe object expression with the letter "X".
  - **Remove.** Remove any information that matches the safe object expression.

- **Regular Expression.** Enter a PCRE-compatible regular expression that defines the safe object. You can create the regular expression in one of three ways: by typing the regular expression directly into the text box, by using the **Regex Tokens** menu to enter regular expression elements and symbols directly into the text box, or by opening the Regular Expressions Editor and using it to construct the expression. The regular expression must consist of ASCII characters only. Do not cut and paste characters that are not part of the basic 128-character ASCII set. If you want to include non-ASCII characters, you must manually type those characters in PCRE hexadecimal character encoding format.
  Note: Do not use start anchors (^) at the beginning of Safe Object expressions, or end anchors ($) at the end of Safe Object expressions. These PCRE entities are not supported in Safe Object expressions, and if used, will cause your expression not to match what it was intended to match.
- **Maximum Match Length.** Enter a positive integer that represents the maximum length of the string that you want to match. For example, if you want to match U.S. social security numbers, enter the number eleven (11) in this field. That allows your regular expression to match a string with nine numerals and two hyphens. If you want to match California driver's license numbers, enter the number eight (8).

  > **Caution**:
  >
  > If you do not enter a maximum match length in this field, the App Firewall uses a default value of one (1) when filtering for strings that match your safe object expressions. As a result, most safe object expressions fail to match their target strings.

You cannot use the command-line interface to configure the Safe Object check. You must configure it by using either the App Firewall wizard or the GUI.

Following are examples of Safe Object check regular expressions:

- Look for strings that appear to be U.S. social security numbers, which consist of three numerals (the first of which must not be zero), followed by a hyphen, followed by two more numerals, followed by a second hyphen, and ending with a string of four more numerals:

```
[1-9][0-9]{
3,3 }
-[0-9]{
2,2 }
-[0-9]{
4,4 }

```

```
<!--NeedCopy-->
```

- Look for strings that appear to be California driver's license IDs, which start with a letter and are followed by a string of exactly seven numerals:

```
[A-Za-z][0-9]{
7,7 }

<!--NeedCopy-->
```

- Look for strings that appear to be Example Manufacturing customer IDs which, consist of a string of five hexadecimal characters (all the numerals and the letters A through F), followed by a hyphen, followed by a three-letter code, followed by a second hyphen, and ending with a string of ten numerals:

```
[0-9A-Fa-f]{
5,5 }
-[A-Za-z]{
3,3 }
-[0-9]{
10,10 }

<!--NeedCopy-->
```

> **Caution**: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write to ensure that they define exactly the type of string you want to add as a safe object definition, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you did not want or expect, such as blocking access to web content that you did not intend to block.

Advanced form protection checks

September 21, 2020

The advanced Form Protection checks examine web form data to prevent attackers from compromising your system by modifying the web forms on your web sites or sending unexpected types and quantities of data to your web site in a form.

> **Note**
>
> SQL, XSS, FFC, and FieldFormat protection checks are applied if the **Exclude Upload Files From Security Checks** is unset.

A file upload is also a form element that has control-name **name** field that is submitted as part of form submission.

Refer to this page, for more information: Forms

Field formats check

September 21, 2020

The Field Formats check verifies the data that users send to your web sites in web forms. It examines both the length and type of data to ensure that it is appropriate for the form field in which it appears. If the App Firewall detects inappropriate web form data in a user request, it blocks the request.

By preventing an attacker from sending inappropriate web form data to your web site, the Field Formats check prevents certain types of attacks on your web site and database servers. For example, if a particular field expects the user to enter a phone number, the Field Formats check examines the user-submitted input to ensure that the data matches the format of a phone number. If a particular field expects a first name, the Field Formats check ensures that the data in that field is of a type and length appropriate for a first name. It does the same thing for each form field that you configure it to protect.

This check applies to HTML requests only. It does not apply to XML requests. You can configure Field Format Checks in HTML profiles or Web 2.0 profiles to inspect HTML payload for protecting your applications. The App Firewall also supports Field Format Check protection for Google Web Toolkit (GWT) applications.

The Field Formats check requires that you enable one or more actions. The App Firewall examines the submitted inputs and applies the specified actions.

> **Note**
>
> Field format rules are tightening rules. Adding them to relaxation list from learned data acts as a blocking rule.
>
> To relax field format rules, please remove particular "fieldname" from the fieldformat relaxations list.

You have the option to set the default field formats to specify Field Type and the minimum and maximum length of data expected in each form field on each web form that you want to protect. You can deploy relaxation rules to configure a Field Format for an individual field of a specific form. Multiple rules can be added to specify the field name, the action URL, and Field Formats. Specify Field Formats to accept different types of inputs in different form fields. The learning feature can provide recommendations for the relaxation rules.

Field Format Actions—You can enable Block, Log, Stats, and Learn actions. At least one of these actions must be enabled to engage the Field Format Check protection.

- **Block**. If you enable block, the block action is triggered if the input does not conform to the specified Field Format. If a rule was configured for the target field, the input is checked against the specified rule. Otherwise, it is checked against the default field format specification. Any mismatch in the Field Type or min/max length specification results in blocking the request.
- **Log**. If you enable the log feature, the Field Format check generates log messages indicating the actions that it takes. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate malicious attempts to launch an attack.
- **Stats**. If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack, or you might have to revisit the configuration to see if the specified field format is too restrictive.
- **Learn**. If you are not sure which Field Types or minimum and maximum length values might be ideally suited for your application, you can use the learn feature to generate recommendations based on the learned data. The App Firewall learning engine monitors the traffic and provides field format recommendations based on the observed values. To get optimal benefit without compromising performance, you might want to enable the learn option for a short time to get a representative sample of the rules, and then deploy the rules and disable learning.
  Note: The App Firewall's learning engine can distinguish only the first 128 bytes of the name. If a form has multiple fields with names that match for the first 128 bytes, the learning engine might not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Default Field Format—In addition to configuring the actions, you can configure the default Field Format to specify the type of data expected in all the form fields for your application. A Field Type can be selected as the Field Format type. Minimum length and Maximum length parameters can be used to specify the length of the allowed inputs. As an alternative to Field Types, you can use Character Maps to specify what's allowed in a field (except in cluster deployments).

- **Field Type**—Field Types are named expression to which you assign assigned priority values. Field Type expressions specify the allowed inputs and are matched against the submitted data to determine whether the received values are consistent with the allowed values. The Field Types are checked in the order of their priority numbers. A lower number indicates a higher priority. The App Firewall gives you the option to add your own Field Types and assign them the priorities you want. The priority value can range from 0 through 64000. The following built-in Field Types are provided to help simplify the configuration process:

```
> sh appfw fieldtype
1)      Name:  integer           Regex:  "^[+-]?[0-9]+$"
                Priority:  30            Comment:  Integer
                Builtin:  IMMUTABLE
2)      Name:  alpha             Regex:  "^[a-zA-Z]+$"
                Priority:  40            Comment:  "Alpha
                   characters"
             Builtin:  IMMUTABLE
3)      Name:  alphanum          Regex:  "^[a-zA-Z0-9]+$"
                Priority:  50            Comment:  "Alpha-numeric
                    characters"
                Builtin:  IMMUTABLE
4)      Name:  nohtml            Regex:  "^[^&<>]*$"
                Priority:  60            Comment:  "Not HTML"
                Builtin:  IMMUTABLE
5)      Name:  any               Regex:  "^.*$"
             Priority:  70            Comment:  Anything
                Builtin:  IMMUTABLE
    Done
>
<!--NeedCopy-->
```

> **Note**: The built-in Field Types are IMMUTABLE. They cannot be modified or removed. Any Field Types that you add are MODIFIABLE. You can edit them or remove them.

Configuring a Field Type as a default Field Format might be useful when you have a PCRE expression that can identify the valid inputs in all or most of the form fields for your application and exclude the invalid inputs. For example, if all the inputs in your application forms are expected to contain only numbers and letters, you might want to use the built-in Field Type alphanum as the default Field Type. Any non-alphanumeric character such as a backslash () or semicolon (;) in the input will trigger a violation. You can also add your own customized Field Types and use them to configure default Field Formats. For example, if you want to make the lowercase "x", "y", and "z" the only allowed alpha characters, you can configure a customized Field Type with regular expression "^[x-z]+$". You could assign it a higher priority (lower priority number) then the built-in Field Types and use it as the default Field Type.

- **Minimum Length** — The default minimum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 0 by default, which allows the user to leave the field blank. Any higher setting forces users to fill in the field.

  Caution: If the minimum length value is 0 but the Field Type is integer, alpha, or alphanum, a request is blocked if any input field is left empty, despite the minimum length setting. That is because the regEx for these Field Types contains a + character, which means one or more characters. Distinguishing an integer from an alpha character requires at least one character.

- **Maximum Length**—The default maximum data length assigned to form fields in web forms that do not have an explicit setting. This parameter is set to 65535 by default.

  > **Note**: Characters vs bytes. The minimum and the maximum lengths for the field formats represent the number of bytes, not the number of characters. Languages that have greater than one-byte character representation can cause the limit to be exceeded with fewer characters than the number configured for the maximum value. For example, with double-byte character representation, the maximum value of 9 allows no more than 4 characters.
  > **Tip**: The GUI allows you to cut and paste UTF-8 characters directly into the GUI without having to convert them to hex.

- **Character Maps**: In addition to recommending the Field Types, the App Firewall learning engine offers you an additional option, Use Character Maps, to deploy the Format Check rules. A Character Map is a set of all the characters allowed in a particular form field. You can fine tune the Field format specification to allow or disallow specific characters by using Character Maps. A separate Character Map is generated for each form field. The alpha and numeric characters are treated differently in Character Maps. If any alpha character is seen in the input, all alpha characters [A-Za-z] will be allowed by the recommended PCRE expression in the Character Map. Similarly, if any digit is included, all digits [0-9] will be allowed. Non-printable characters are specified by using the x construct. Only single Byte characters with values between 0-255 are considered for Character Map recommendations.

  A Character Map can be more specific than the corresponding Field Type recommendation. In some situations, Character Maps might be a better option, because they give you tighter control over the set of characters allowed as inputs. The deployed character maps are displayed as strings that start with prefix "CM" followed by digits. The priority for the Character Maps starts at 10000. As with user-added Field Types, you can add, edit or remove a Character Map. Character Maps that are currently used in deployed rules cannot be modified or removed.

  Note: Character Maps are not supported in cluster deployments.

> **Note**
>
> When you add a field formats rule with any built-in Field Type and use character map instead of Field Type and save it, the changes do not get saved and rule still shows with Field Type.
>
> When the character map matches one of the built-in type, the field type is reused instead of creating a new character map.

Using the command line to configure the field format check

In the command line interface, you can use the add appfw fieldtype command to add a new Field Type. You can use either the set appfw profile command or the add appfw profile command to configure the Field Format check and specify which actions to perform. You can use the unset appfw profile command to revert the configured settings back to their defaults. To specify a Field Format rule, use the bind appfw command to bind a Field Type to a form Field and the action URL, along with the minimum and maximum length specifications.

To add, remove or view a Field Type by using the command line:

Use the add command to add a Field Type. You must specify the Name, Regular expression and Priority when adding a new Field Type. You also have the option to add a Comment. You can use the show command to display the configured Field Types. You can also delete a Field Type by using the remove command, which requires only the Name of the Field Type.

`add [appfw] fieldType <name> <regex> <priority> [-comment <string>]`
where:

`<regex>` is a reqular expression

`<priority>` is a positive_integer

Example:

```
add fieldtype "Cust_Zipcode" "^[0-9]{
 5 }
 [-][0-9]{
 4 }
 $" 4

-   show [appfw] fieldType [<name>]

    Example: sh fieldtype

    sh appfw fieldtype

    sh appfw fieldtype cust_zipcode

-   `rm [appfw] fieldType <name>`

    Example: rm fieldtype cusT_ziPcode

    `rm appfw fieldtype cusT_ziPcode`
<!--NeedCopy-->
```

> **Note**: As shown above, use of "appfw" in the command is optional. For example, "Add Field-Type" or "Add appfw fieldType" are both valid options. The names of the Field Types are case insensitive due to normalization. As shown in the above examples, Cust_Zipcode, cust_zipcode, and cUsT_ziPcode refer to the same Field Type.

To configure a Field Format check by using the command line

Use either the set appfw profile command or the add appfw profile command, as follows:

- `set appfw profile <name> -fieldFormatAction (([block] [learn] [log] [ stats])| [none])`
- `set appfw profile <name>-defaultFieldFormatType <string>`
- `set appfw profile <name> -defaultFieldFormatMinLength <integer>`
- `set appfw profile <name> -defaultFieldFormatMaxLength <integer>`

To configure a Field Format relaxation rule by using the command line

```
bind appfw profile <name> (-fieldFormat <string> <formActionURL>  <
    fieldType>
[-fieldFormatMinLength <positive_integer>]  [-fieldFormatMaxLength <
    positive_integer>]
[-isRegex ( REGEX | NOTREGEX )])
<!--NeedCopy-->
```

Example:

```
bind appfw profile pr_ffc -fieldFormat "login_name" ".*/login.php"
    integer -fieldformatMinLength 3 -FieldformatMaxlength 6
<!--NeedCopy-->
```

Using the GUI to Configure the field formats security check

In the GUI, you can manage the Field Types. You can also configure the Field Formats security check in the pane for the profile associated with your application.

To add, modify or remove a Field Type using the GUI

1. Navigate to Application Firewall node. In the Settings, click **Manage Field Types** to display the Configure Application Firewall Field Type dialogue box.
2. Click **Add** to add a new Field Type. Follow the instructions in this pane and click Create. You can also edit or delete any user-added Field Type if it is currently not being used by a deployed rule.

To add or modify the Field Formats security check by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.

2. In the **Advanced Settings** pane, click **Security Checks**.

   The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

   a) If you just want to enable or disable **Block**, **Log**, **Stats**, and **Learn** actions for Field Formats, you can select or clear check boxes in the table, click **OK**, and then click **Save and Close** to close the Security Check pane.

   b) If you want to configure additional options for this security check, double click Field Formats, or select the row and click Action Settings, to display the following options for **Default Field Format**:
      - **Field Type**—Select the Field Type that you want to configure as the default Field Type. You can select the built-in and user-defined Field Types. The deployed Character Maps are also included in the list and can be selected.
      - **Minimum Length**—Specify the minimum number of characters that must be in each field. Possible values: 0-65535.
      - **Maximum Length**— Specify the maximum number of characters that must be in each field. Possible values: 1-65535.
        You can also edit the **Block**, **Log**, **Stats** and **Learn** actions in the Field Formats Settings pane.

   After making any of the above changes, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click OK **** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

- To configure a Field Formats relaxation rule by using the GUI

  1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.

  2. In the **Advanced Settings** pane, click **Relaxation Rules**. The Relaxation Rules table has a Field Formats entry. You can double click, or select this row and click the Edit button, to access the Field Formats Relaxation Rules dialogue. You can perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

     For a consolidated view of all the relaxation rules, you can highlight the Field Formats row and click Visualizer. The visualizer for deployed relaxations offers you the option to Add a new rule or Edit an existing one. You can also Enable or Disable a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

Using the learn feature with the Field Formats Check

When the learn action is enabled, the App Firewall learning engine monitors the traffic and learns the triggered violations. You can periodically inspect these learned rules. After due consideration, you can deploy the learned rule as a Field Format relaxation rule.

Field formats learning enhancement—An App Firewall learning enhancement was introduced in release 11.0. In the previous releases, once the learned field format recommendation is deployed, the App Firewall learning engine stops monitoring the valid requests for the purpose of recommending new rules on the basis of the new data points. This limits the configured security protection, because the learning database does not include any representations of the new data seen in the valid requests processed by the security check.

Violations are no longer coupled with learning. The learning engine learns and makes recommendations for the field formats regardless of the violations. In addition to checking the blocked requests to determine whether the current field format is too restrictive and needs to be relaxed, the learning engine also monitors the allowed requests to determine whether the current field format is too permissive, and allows elevating the security by deploying a more restrictive rule.

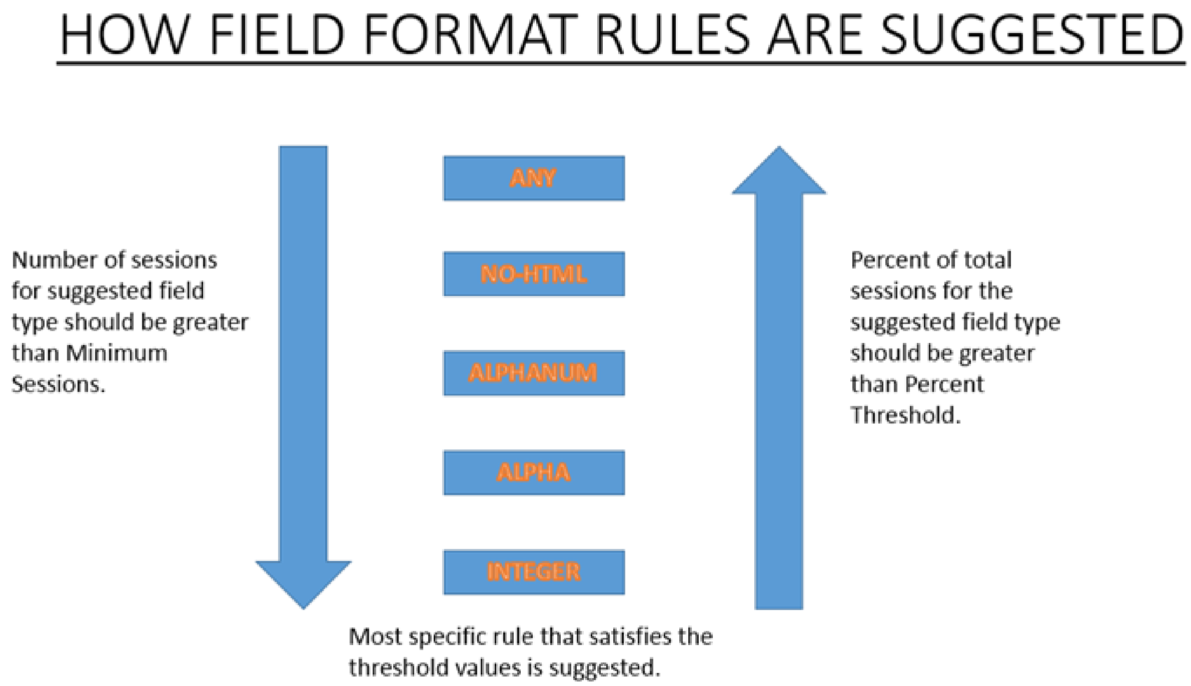
Following is a summary of the Field Formats learning behavior:

No Field format is bound—The behavior remains unchanged in this scenario. All learn data is sent to the aslearn engine. The learning engine suggests a field format rule based on the data set.

Field format is bound: In the previous releases, observed data is sent to the aslearn engine only in the case of a violation. The learning engine suggests a field format rule based on the data set. In the 11.0 release, all data is sent to aslearn engine even if no violation is triggered. The learning engine suggests a field format rule based on the entire data set of all received inputs.

Use Case for learning enhancement:

If the initial field format learned rules are based on a small sample of data, a few non typical values might result in a recommendation that is too lenient for the target field. The ongoing learning allows the App Firewall to observe data points from every request to collect a representative sample for the learned recommendations. This is helpful in further tightening the security to deploy the optimal input format with an adequate range value.


HOW FIELD FORMAT RULES ARE SUGGESTED
Number of sessions for suggested field type should be greater than Minimum Sessions.
ANY
NO-HTML
ALPHANUM
ALPHA
INTEGER
Percent of total sessions for the suggested field type should be greater than Percent Threshold.
Most specific rule that satisfies the threshold values is suggested.

The field format learning makes use of the priority of the Field Types as well as the configured settings of the following learning thresholds:

- **FieldFormatMinThreshold**—Minimum number of times a specific form field must be observed before a learned relaxation is generated. Default: 1.
- **FieldFormatPercentThreshold**—Percentage of times a form field matched a particular Field Type, before a learned relaxation is generated. Default: 0.

The field format rule recommendations are based on the following criteria:

- **Field Type recommendations**—The Field Type recommendations are determined by the assigned priorities of the existing Field Types and the specified Field Format thresholds. The priorities determine the order in which the Field Types are matched against the inputs. A lower number specifies a higher priority. For example, Field Type integer has the higher priority (30) and is therefore evaluated before Field Type alphanum (50). The thresholds determine the number of inputs evaluated to collect a representative sample for the data point. Assigning the right priority to the configured Field Types, and configuring an appropriate **learningsetting** value for the **fieldFormatPercentThreshold** and **fieldFormatMinThreshold** parameters, is essential for getting the correct Field Format recommendation. The Field Type with the highest priority, based on the configured thresholds, is matched first against the inputs. If there is a match, this Field Type is suggested without considering the other Field Types. For example, three default Field Types, integer, alphanum, and any will match if all the inputs contain only numbers. However, integer will be recommended since it has the highest priority.
- **Minimum and Maximum length recommendations**—Calculations for the minimum and maximum lengths for the Field Format are done independently of the determination for the Field

Type. The field format length calculations are based on the average length of all the observed inputs. Half of this calculated average is suggested as the min value, and twice the value of this average is suggested as the max value. The range for the Minimum Length is 0-65535 and the range for the Maximum Length is 1-65535. The configured value for the minimum length cannot exceed the maximum length.

- **Handling of space character**—The Field Format check counts every space character when checking for the Field Formats length. Leading or trailing spaces are not stripped, and multiple consecutive spaces in the middle of the input string are no longer consolidated to a single space during input processing.

Example to illustrate the Field Format recommendations:

```
Total requests: 100
Number of Req with Field Type:
Int : 22            (22 int values)  -  22%
Alpha : 44                  (44 alpha values)  -  44%
Alphanum: 14            (14 + 44 + 22 = 80 alphanum values) = 80%
noHTML: 10              (80 + 10 = 90 noHTML values) = 90%
any : 10                     (90 + 10 = 100 any values) = 100%

% threshold                              Suggested Field Type
0-22                int
23-44               alpha
45-80               alphanum
81-90               noHTML
91-100              any
<!--NeedCopy-->
```

To view or use learned data by using the command line interface

```
show appfw learningdata <profilename> FieldFormat
rm appfw learningdata <profilename> -fieldFormat <string>  <
    formActionURL>
export appfw learningdata <profilename> FieldFormat
<!--NeedCopy-->
```

To view or use learned data by using the GUI

1. Navigate to **Application Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Learned Rules**. You can select the Field Formats entry in the Learned Rules table and double-click it to access the learned rules. You can deploy the learned rules or edit a rule before deploying it as a relaxation rule. To discard a rule, you can select it and click the **Skip** button. You can edit only one rule at a time, but you can select multiple rules to deploy or skip.

You also have the option to show a summarized view of the learned relaxations by selecting the Field Formats entry in the Learned Rules table and clicking Visualizer to get a consolidated view of all the learned violations. The visualizer makes it very easy to manage the learned rules. It presents a comprehensive view of the data on one screen and facilitates taking action on a group of rules with one click. The biggest advantage of the visualizer is that it recommends regular expressions to consolidate multiple rules. You can select a subset of these rules, based on the delimiter and Action URL. You can display 25, 50, or 75 rules in the visualizer, by selecting the number from a drop-down list. The visualizer for learned rules offers the option to edit the rules and deploy them as relaxations. Or you can skip the rules to ignore them.

Using the log feature with the field formats check

When the log action is enabled, the Field Formats security check violations are logged in the audit log as APPFW_FIELDFORMAT violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the `/var/log/` folder to access the log messages pertaining to the Field Formats violations:

- `Shell`
- `tail -f /var/log/ns.log | grep APPFW_FIELDFORMAT`

To access the log messages by using the GUI

The NetScaler GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **Application Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the Field Formats row and click **Logs**. When you access the logs directly from the **Field Formats security** check of the profile, it filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the **Syslog Viewer**, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **Application Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

  The HTML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To access Field Formats security check violation log messages, filter by selecting APPFW in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the **APPFW_FIELDFORMAT** check box and click the **Apply** button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as Module and EventType, appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Example of a Native format log message when the request is not blocked

```
Jun 10 22:32:26 <local0.info> 10.217.31.98 06/10/2015:22:32:26 GMT ns0-PPE-0 :
default APPFW APPFW_FIELDFORMAT 97 0 :  10.217.253.62 562-PPE0
x1MV+YnNGzQFM3Bsy2wti4bhXio0001 pr_ffc http://aaron.stratum8.net/FFC/login_post.php
Field format check failed for field passwd="65568888sz-*_" <not blocked>
Example of a CEF format log message when the request is blocked
Jun 11 00:03:51 <local0.info> 10.217.31.98
CEF:0|Citrix|NetScaler|NS11.0|APPFW|APPFW_FIELDFORMAT|6|src=10.217.253.62 spt=27076
method=POST requet=http://aaron.stratum8.net/FFC/maxlen_post.php msg=Field format check
failed for field text_area="" cn1=108 cn2=644 cs1=pr_ffc cs2=PPE0
cs3=GaUROfl1Nx1jJTvja5twH5BBqI0000 cs4=ALERT cs5=2015 act=blocked
<!--NeedCopy-->
```

Statistics for the field formats violations

When the stats action is enabled, the corresponding counter for the Field Formats check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, the request for a page that contains 3 Field Format violations increments the stats counter by one, because the page is blocked as soon as the first Field Formats violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by 3, because each Field Formats violation generates a separate log message.

To display Field Formats statistics by using command line

At the command prompt, type:

`sh appfw stats`

To display stats for a specific profile, use the following command:

```
stat appfw profile <profile name>
```

To display Field Formats statistics by using GUI

1. Navigate to **System** > **Security** > **Application Firewall**.
2. In the right pane, access the Statistics Link.
3. Use the scroll bar to view the statistics about Field Formats violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

Deployment tip

- Enable Field format actions log, learn and stats.
- After running a representative sample of the traffic to your application, review the learned recommendations.
- If a Field Type is recommended by most of the learned rules, configure that Field Type as the Default Field Type. For minimum and maximum lengths, use the widest range suggested by these rules.
- Deploy rules for other fields for which different Field Types or different minimum/maximum lengths are better suited.
- Enable blocking and disable learning.
- Monitor stats and logs. If a significant number of violations are still being triggered, you might want to review the log messages to confirm that the violations represent malicious requests that should have been blocked. If valid requests are being flagged as violations, you can either edit the configured Field Format rule to further relax it or enable learning again to get recommendations based on the new data points.

> **Note**: You can fine tune your configuration by getting new learning recommendations.

Highlights

Note the following points about the Field Format security check:

- **Protection**—By configuring optimal field format rules, you can protect against many attacks. For example, if you specify that a field can only have integers, hackers will not be able to launch SQL Injection or XSS attacks by using this field, because the inputs required to launch such attacks will not meet the configured field format requirement.
- **Performance**—You can limit the minimum and the maximum allowed length for the inputs in the field format rules. This can prevent a malicious user from entering excessively large input strings in an attempt to add processing overhead to the server, or worse, cause the server to dump core because of stack overflow. By limiting the input size, you can shorten the time required for processing legitimate requests.

- **Configuring Field Formats**—You must enable one of the actions (block, log, stats, learn) to engage the field Format protection. You can also specify the Field format rules to identify the allowed inputs in your form fields.
- **Selecting Character Maps vs. Field Types**—Both Character Maps and Field Types use regular expressions. However, a Character Map provides a more specific expression by narrowing down the list of allowed characters. For example, for an input such as janedoe@citrix.com, the learning engine might recommend the Field Type nohtml but the Character Map [.@-Za-z] might be more specific, because it narrows down the allowed set of non-alpha characters. The Character Map option allows, in addition to alpha characters, only two non-alpha characters: period (.) and at (@).
- **Ongoing Learning**—The App Firewall monitors and takes into account all the incoming data (violations as well as allowed inputs) to build a learning table for recommending rules. The rules are revised and updated as new incoming data arrives. New field format rules are suggested for a field even if it already has a bound field format rule. If the configured Field Formats are too restrictive and are blocking the valid requests, you can deploy a more relaxed Field Format. Similarly, if the current Field Formats are too generic, you can further refine and tighten the security by deploying a more restrictive Field Format.
- **Overwriting Rules**—If a rule has already been deployed for a field/URL combination, the GUI allows the user to update the field format. A dialog box asks for confirmation to replace the existing rule. If you are using the command line interface, you have to explicitly unbind the previous binding and then bind the new rule.
- **Multiple match**—If multiple field formats match a given field name and its action URL, the App Firewall arbitrarily selects one of them to apply.
- **Buffer boundary**—If a field value extends across multiple streaming buffers, and the format for these two parts of the field value is different, a field format corresponding to “any” is sent to the learn database.
- **Field Format vs. Field Consistency Check**—Both the Field Format check and the Field Consistency check are form-based protection checks. The Field Formats check provides a different type of protection than does the Form Field Consistency check. The Form Field Consistency check verifies that the structure of the web forms returned by users is intact, that data format restrictions configured in the HTML are respected, and that data in hidden fields has not been modified. It can do this without any specific knowledge about your web forms other than what it derives from the web form itself. The Field Formats check verifies that the data in each form field matches the specific formatting restrictions that you configured manually, or that the learning feature generated and you approved. In other words, the Form Field Consistency check enforces general web form security, while the Field Formats check enforces the specific rules for the allowed inputs for your web forms.

Form field consistency check

September 21, 2020

The Form Field Consistency check examines the web forms returned by users of your web site, and verifies that web forms were not modified inappropriately by the client. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The Form Field Consistency check prevents clients from making unauthorized changes to the structure of the web forms on your web site when they fill out and submit a form. It also ensures that the data a user submits meets the HTML restrictions for length and type, and that data in hidden fields is not modified. This prevents an attacker from tampering with a web form and using the modified form to gain unauthorized access to web site, redirect the output of a contact form that uses an insecure script and thereby send unsolicited bulk email, or exploit a vulnerability in your web server software to gain control of the web server or the underlying operating system. Web forms are a weak link on many web sites and attract a wide range of attacks.

The Form Field Consistency check verifies all of the following:

- If a field is sent to the user, the check ensures that it is returned by the user.
- The check enforces HTML field lengths and types.

  > **Note**: The Form Field Consistency check enforces HTML restrictions on data type and length but does not otherwise validate the data in web forms. You can use the Field Formats check to set up rules that validate data returned in specific form fields on your web forms.

- If your web server does not send a field to the user, the check does not allow the user to add that field and return data in it.
- If a field is a read-only or hidden field, the check verifies that the data has not changed.
- If a field is a list box or radio button field, the check verifies that the data in the response corresponds to one of the values in that field.

If a web form returned by a user violates one or more of the Form Field consistency checks, and you have not configured the App Firewall to allow that web form to violate the Form Field Consistency checks, the request is blocked.

If you use the wizard or the GUI, in the Modify Form Field Consistency Check dialog box, on the General tab you can enable or disable the Block, Log, Learn, and Statistics actions.

You also configure Sessionless Field Consistency in the General tab. If Sessionless Field Consistency is enabled, the App Firewall checks only the web form structure, dispensing with those parts of the Form Field Consistency check that depend upon maintaining session information. This can speed the Form Field Consistency check with little security penalty for web sites that use many forms. To use

Sessionless Field Consistency on all web forms, select On. To use it only for forms submitted with the HTTP POST method, select postOnly

If you use the command-line interface, you can enter the following command to configure the Form Field Consistency Check:

- `set appfw profile <name> -fieldConsistencyAction [block] [learn] [log] [stats] [none]`

To specify relaxations for the Form Field Consistency check, you must use the GUI. On the Checks tab of the Modify Form Field Consistency Check dialog box, click Add to open the Add Form Field Consistency Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Form Field Consistency Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation, as described in "Manual Configuration By Using the GUI."

Following are examples of Form Field Consistency check relaxations:

Form Field Names:

- Choose form fields with the name UserType:

  ```
  ^UserType$
  <!--NeedCopy-->
  ```

- Choose form fields with names that begin with UserType_ and are followed by a string that begins with a letter or number and consists of from one to twenty-one letters, numbers, or the apostrophe or hyphen symbol:

  ```
  ^UserType_[0-9A-Za-z][0-9A-Za-z'-]{
  0,20 }
  $
  <!--NeedCopy-->
  ```

- Choose form fields with names that begin with Turkish-UserType_ and are otherwise the same as the previous expression, except that they can contain Turkish special characters throughout:

  ```
  ^T\xC3\xBCrk\xC3\xA7e-UserType_([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])+$
  <!--NeedCopy-->
  ```

  > **Note**: See "PCRE Character Encoding Format" for a complete description of supported special characters and how to encode them properly.

- Choose form field names that begin with a letter or number, consist of a combination of letters and/or numbers only, and that contain the string Num anywhere in the string:

  ```
  ^[0-9A-Za-z]*Num[0-9A-Za-z]*$
  ```

```
<!--NeedCopy-->
```

Form Field Action URLs:

- Choose URLs beginning with http://www.example.com/search.pl? and containing any string after the query except for a new query:

```
^http://www[.]example[.]com/search[.]pl?[^?]*$
<!--NeedCopy-->
```

- Choose URLs that begin with http://www.example-español.com and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
^http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
<!--NeedCopy-->
```

> **Caution**: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (
> .*) metacharacter/wildcard combination, can have results you do not want or expect, such as blocking access to web content that you did not intend to block or allowing an attack that the Cookie Consistency check would otherwise have blocked.

CSRF form tagging check

September 21, 2020

The Cross Site Request Forgery (CSRF) Form Tagging check tags each web form sent by a protected web site to users with a unique and unpredictable FormID, and then examines the web forms returned by users to ensure that the supplied FormID is correct. This check protects against cross-site request forgery attacks. This check applies only to HTML requests that contain a web form, with or without data. It does not apply to XML requests.

The CSRF Form Tagging check prevents attackers from using their own web forms to send high volume form responses with data to your protected web sites. This check requires relatively little CPU processing capacity compared to certain other security checks that analyze web forms in depth. It is therefore able to handle high volume attacks without seriously degrading the performance of the protected web site or the App Firewall itself.

Before you enable the CSRF Form Tagging check, you should be aware of the following:

- You need to enable form tagging. The CSRF check depends on form tagging and does not work without it.
- You should disable the NetScaler Integrated Caching feature for all web pages containing forms that are protected by that profile. The Integrated Caching feature and CSRF form tagging are not compatible.
- You should consider enabling Referer checking. Referer checking is part of the Start URL check, but it prevents cross-site request forgeries, not Start URL violations. Referer checking also puts less load on the CPU than does the CSRF Form Tagging check. If a request violates Referer checking, it is immediately blocked, so the CSRF Form Tagging check is not invoked.
- The CSRF Form Tagging check does not work with web forms that use different domains in the form-origin URL and form-action URL. For example, CSRF Form Tagging cannot protect a web form with a form-origin URL of `http://www.example.com` and a form action URL of `http://www.example.org/form`.pl, because `example.com` and `example.org` are different domains.

If you use the wizard or the GUI, in the **Modify CSRF Form Tagging Chec**k dialog box, on the **General** tab you can enable or disable the Block, Log, Learn and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the CSRF Form Tagging Check:

- `set appfw profile <name> -CSRFtagAction [block] [log] [learn] [stats] [none]`

To specify relaxations for the CSRF Form Tagging check, you must use the GUI. On the Checks tab of the **Modify CSRF Form Tagging Check** dialog box, click **Add** to open the **Add CSRF Form Tagging Check Relaxation** dialog box, or select an existing relaxation and click **Open** to open the **Modify CSRF Form Tagging Check Relaxation** dialog box. Either dialog box provides the same options for configuring a relaxation.

An alert is generated when you set the NetScaler Appfirewall session limit to a value of 0 or lower, because such a setting affects advanced protection check functionality that requires a properly functioning App Firewall session.

Following are examples of CSRF Form Tagging check relaxations:

Note: The following expressions are URL expressions that can be used in both the Form Origin URL and Form Action URL roles.

- Choose URLs beginning with `http://www.example.com/search.pl?` and containing any string after the query, except for a new query:

```
^http://www[.]example[.]com/search
[.]pl?[^?]*$
<!--NeedCopy-->
```

- Choose URLs that begin with `http://www.example-español.com` and have paths and filenames that consist of upper-case and lower-case letters, numbers, non-ASCII special characters, and selected symbols in the path. The ñ character and any other special characters are represented as encoded UTF-8 strings containing the hexadecimal code assigned to each special character in the UTF-8 charset:

```
http://www[.]example-espa\xC3\xB1ol[.]com/(([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])
([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|\\x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|\\x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Choose all URLs that contain the string /search.cgi?:

```
^[^?<>]*/search[.]cgi\?[^?<>]*$
<!--NeedCopy-->
```

> **Important**
>
> Regular expressions are powerful. If you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL that you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the check would otherwise have blocked.
>
> **Tip**
>
> When enableValidate referrer header is enabled under the Start URL Action, ensure that the Referrer Header URL is added to StartURL as well.
>
> **Note**
>
> When NetScaler reaches the `appfw_session_limit` and CSRF checks are enabled, the web application freezes.

> To prevent web application freeze, decrease the session timeout and increase the session limit by using the following commands:

```
> From CLI: > set appfw settings -sessiontimeout 300
> From shell: root@ns# nsapimgr_wr.sh -s appfw_session_limit=200000
<!--NeedCopy-->
```

> Logging and generating SNMP alarms when appfw_session_limit is reached assists you in troubleshooting and debugging issues.

Managing CSRF form tagging check relaxations

September 22, 2020

You configure an exception (or relaxation) to the CSRF Form Tagging security check in the Add Cross-Site Request Forgery Tagging Check Relaxation dialog box or the Modify Cross-Site Request Forgery Tagging Check Relaxation dialog box.

To configure a CSRF form tagging check relaxation by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the **Profiles** pane, select the profile you want to configure, and then click **Open**.
3. In the **Configure App Firewall Profile** dialog box, click the **Security Checks** tab. The **Security Checks** tab contains the list of App Firewall security checks.
4. To add or modify a CSRF relaxation, do one of the following:
    - To add a new relaxation, click Add.
    - To modify an existing relaxation, select the relaxation that you want to modify, and then click **Open**.

   The **Add Cross-Site Request Forgery Tagging Check Relaxation** or **Modify Cross-Site Request Forgery Tagging Check Relaxation** dialog box is displayed. Except for the title, these dialog boxes are identical.
5. Fill in the dialog box as described below.
    - **Enabled check box**—Select to place this relaxation or rule in active use; clear to deactivate it.
    - **Form Origin URL**—In the text area, enter a PCRE-format regular expression that defines the URL that hosts the form.

   - **Form Action URL**—In the text area, enter a PCRE-format regular expression that defines the URL to which data entered into the form is delivered.
   - **Comments**—In the text area, type a comment. Optional.

> **Note**: For any element that requires a regular expression, you can type the regular expression, use the **Regex Tokens** menu to insert regular expression elements and symbols directly into the text box, or click **Regex Editor** to open the **Add Regular Expression** dialog box, and use it to construct the expression.

6. Click **OK**. The **Add Cross-Site Request Forgery Tagging Check Relaxation** or **Modify Cross-Site Request Forgery Tagging Check Relaxation** dialog box closes and you return to the **Modify Cross-Site Request Forgery Tagging Check** dialog box.
7. To remove a relaxation or rule, select it, and then click **Remove**.
8. To enable a relaxation or rule, select it, and then click **Enable**.
9. To disable a relaxation or rule, select it, and then click **Disable**.
10. To configure the settings and relationships of all existing relaxations in an integrated interactive graphic display, click **Visualizer**, and use the display tools.
11. To review and configure learned rules for the CSRF check, click **Learning**and perform the steps in "To configure and use the Learning feature."
12. Click **OK**.

URL protection checks

September 21, 2020

The URL Protection checks examine request URLs to prevent attackers from aggressively attempting to access multiple URLs (forceful browsing) or using a URL to trigger a known security vulnerability in web server software or web site scripts.

Start URL check

September 21, 2020

The Start URL check examines the URLs in incoming requests and blocks the connection attempt if the URL does not meet the specified criteria. To meet the criteria, the URL must match an entry in the Start URL list, unless the Enforce URL Closure parameter is enabled. If you enable this parameter, a user who clicks a link on your Web site is connected to the target of that link.

The primary purpose of the Start URL check is to prevent repeated attempts to access random URLs on a Web site, (forceful browsing) through bookmarks, external links, or jumping to pages by manually typing in the URLs to skip the pages required to reach that part of the website. Forceful browsing can be used to trigger a buffer overflow, find content that users were not intended to access directly, or find a back door into secure areas of your Web server. The App Firewall enforces a website's given traversal or logic path by allowing access to only the URL's that are configured as start URLs.

If you use the wizard or the GUI, in the Modify Start URL Check dialog box, on the
General tab you can enable or disable Block, Log, Statistics, Learn actions, and the following parameters:

- **Enforce URL Closure.** Allow users to access any web page on your web site by clicking a hyperlink on any other page on your web site. Users can navigate to any page on your web site that can be reached from the home page or any designated start page by clicking hyperlinks.
  Note: The URL closure feature allows any query string to be appended to and sent with the action URL of a web form submitted by using the HTTP GET method. If your protected web sites use forms to access an SQL database, make sure that you have the SQL injection check enabled and properly configured.
- **Sessionless URL Closure.** From the client's point of view, this type of URL closure functions in exactly the same way as standard, session-aware URL Closure, but uses a token embedded in the URL instead of a cookie to track the user's activity, which consumes considerably fewer resources. When sessionless URL closure is enabled, the App Firewall appends a "as_url_id" tag to all the URL's that are in URL closure.
  Note: When enabling sessionless (Sessionless URL Closure), you must also enable regular URL closure (Enforce URL Closure) or sessionless URL closure does not work.
- **Validate Referer Header.** Verify that the Referer header in a request that contains web form data from your protected web site instead of another web site. This action verifies that your web site, not an outside attacker, is the source of the web form. Doing so protects against cross-site request forgeries (CSRF) without requiring form tagging, which is more CPU-intensive than header checks. The App Firewall can handle the HTTP Referer header in one of the following four ways, depending on which option you select in the drop-down list:
  - **Off**—Do not validate the Referer header.
  - **If-Present**—Validate the Referer header if a Referer header exists. If an invalid Referer header is found, the request generates a referer-header violation. If no Referer header exists, the request does not generate a referer-header violation. This option enables the App Firewall to perform Referer header validation on requests that contain a Referer header, but not block requests from users whose browsers do not set the Referer header or who use web proxies or filters that remove that header.
  - **Always Except Start URLs**—Always validate the Referer header. If there is no Referer header and the requested URL is not exempted by the startURL relaxation rule, the request generates a referer-header violation. If the Referer header is present but it is invalid, the

request generates a referer-header violation.

  - **Always Except First Request**—Always validate the referer header. If there is no referer header, only the URL that is accessed first is allowed. All other URL's are blocked without a valid referer header. If the Referer header is present but it is invalid, the request generates a referer-header violation.

One Start URL setting, **Exempt Closure URLs from Security Checks**, is not configured in the Modify Start URL Check dialog box, but is configured in the Settings tab of the Profile. If enabled, this setting directs the App Firewall not to run further form based checks (such as Cross-Site Scripting and SQL Injection inspection) on URLs that meet the URL Closure criteria.

> **Note**
>
> Although the referer header check and Start URL security check share the same action settings, it is possible to violate the referer header check without violating the Start URL check. The difference is visible in the logs, which log referer header check violations separately from Start URL check violations.

The Referer header settings (OFF, if-Present, AlwaysExceptStartURLs, and AlwaysExceptFirstRequest) are arranged in order of least restrictive to most restrictive and work as follows:

OFF:

- Referer Header Not checked.

If-Present:

- Request has no referer header -> Request is allowed.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptStartURLs:

- Request has no referer header and the request URL is a start URL -> Request is allowed.
- Request has no referer header and the request URL is not a start URL ->Request is blocked.
- Request has referer header and the referer URL is in URL closure -> Request is allowed.
- Request has referer header and the referer URL is **not** in URL closure -> Request is blocked.

AlwaysExceptFirstRequest:

- Request has no referer header and is the first request URL of the session -> Request is allowed.
- Request has no referer header and is **not** the first request URL of the session -> Request is blocked.
- Request has referer header and is either the first request URL of the session or is in URL closure -> Request is allowed.
- Request has referer header and is neither the first request URL of the session nor is in URL closure -> Request is blocked.

If you use the command-line interface, you can enter the following commands to configure the Start URL Check:

- `set appfw profile <name> -startURLAction [block] [learn] [log] [stats] [none]`
- `set appfw profile <name> -startURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -sessionlessURLClosure ([ON] | [OFF])`
- `set appfw profile <name> -exemptClosureURLsFromSecurityChecks ([ON] | [OFF)`
- `set appfw profile <name> -RefererHeaderCheck ([OFF] | [if-present] | [AlwaysExceptStartURLs] | [AlwaysExceptFirstRequest])`

To specify relaxations for the Start URL check, you must use the GUI. On the Checks tab of the Modify Start URL Check dialog box, click Add to open the Add Start URL Check Relaxation dialog box, or select an existing relaxation and click Open to open the Modify Start URL Check Relaxation dialog box. Either dialog box provides the same options for configuring a relaxation.

Following are examples of Start URL check relaxations:

- Allow users to access the home page at www.example.com:

```
    ^http://www[.]example[.]com$
<!--NeedCopy-->
```

- Allow users to access all static HTML (.htm and .html), server-parsed HTML (.htp and .shtml), PHP (.php), and Microsoft ASP (.asp) format web pages at www.example.com:

```
    ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
    [0-9A-Za-z][0-9A-Za-z_.-]*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

- Allow users to access web pages with pathnames or file names that contain non-ASCII characters at www.example-español.com:

```
    ^http://www[.]example-espaxC3xB1ol[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*/)*
    ([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](asp|htp|php|s?html?)$
<!--NeedCopy-->
```

```
**Note**: In the above expression, each character class has been grouped with the string
x[0-9A-Fa-f][0-9A-Fa-f], which matches all properly-constructed character encoding strings but does not allow stray backslash characters that are not associated with a UTF-8 character encoding
```

```
    string. The double backslash () is an escaped backslash, which tells
     the App Firewall to interpret it as a literal backslash. If you
    included only one backslash, the App Firewall would instead
    interpret the following left square bracket ([) as a literal
    character instead of the opening of a character class, which would
    break the expression.
```

- Allow users to access all GIF (.png), JPEG (.jpg and .jpeg), and PNG (.png) format graphics at www.example.com:

```
    ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
    [0-9A-Za-z][0-9A-Za-z_.-]*[.](gif|jpe?g|png)$
<!--NeedCopy-->
```

- Allow users to access CGI (.cgi) and PERL (.pl) scripts, but only in the CGI-BIN directory:

```
    ^http://www[.]example[.]com/CGI-BIN/[0-9A-Za-z][0-9A-Za-z_.-]*[.](
        cgi|pl)$
<!--NeedCopy-->
```

- Allow users to access Microsoft Office and other document files in the docsarchive directory:

```
    ^http://www[.]example[.]com/docsarchive/[0-9A-Za-z][0-9A-Za-z_
        -.]*[.](doc|xls|pdf|ppt)$
<!--NeedCopy-->
```

> **Note**
>
> By default, all App Firewall URLs are considered to be regular expressions.

Caution: Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions that you write. Make sure that they define exactly the URL you want to add as an exception, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (.*) metacharacter/wildcard combination, can have results you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the Start URL check would otherwise have blocked.

> **Tip**
>
> You can add the *-and-* to the allowed list of SQL keywords for URL naming scheme. For example, example https://FQDN/bread-and-butter.

Deny URL check

September 21, 2020

The Deny URL check examines and blocks connections to URLs that are commonly accessed by hackers and malicious code. This check contains a list of URLs that are common targets of hackers or malicious code and that rarely if ever appear in legitimate requests. You can also add URLs or URL patterns to the list. The Deny URL check prevents attacks against various security weaknesses known to exist in web server software or on many web sites.

The Deny URL check takes priority over the Start URL check, and thus denies malicious connection attempts even when a Start URL relaxation would normally allow a request to proceed.

In the Modify Deny URL Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the Deny URL Check:

- `set appfw profile <name> -denyURLAction [block] [log] [stats] [none]`

To create and configure your own deny URLs, you must use the GUI. On the Checks tab of the Modify Deny URL Check dialog box, click Add to open the Add Deny URL dialog box, or select an existing user-defined deny URL and click Open to open the Modify Deny URLdialog box. Either dialog box provides the same options for creating and configuring a deny URL.

Following are examples of Deny URL expressions:

- Do not allow users to access the image server at images.example.com directly:

```
    ^http://images[.]example[.]com$
<!--NeedCopy-->
```

- Do not allow users to access CGI (.cgi) or PERL (.pl) scripts directly:

```
    ^http://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-]*/)*
    [0-9A-Za-z][0-9A-Za-z_.-]*[.](cgi|pl)$
<!--NeedCopy-->
```

- Here is the same deny URL, modified to support non-ASCII characters:

```
    ^http://www[.]example[.]com/(([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])
    ([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*/)*([0-9A-Za-z]|x[0-9A-Fa-f][0-9A-Fa-f])
    ([0-9A-Za-z_-]|x[0-9A-Fa-f][0-9A-Fa-f])*[.](cgi|pl)$
<!--NeedCopy-->
```

> **Caution**:
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the URL or pattern that you want to block, and nothing else. Careless use of wildcards, and especially of the dot-asterisk (
> .*) metacharacter/wildcard combination, can have results that you do not want, such as blocking access to web content that you did not intend to block.

XML protection checks

September 21, 2020

The XML Protection checks examine requests for XML-based attacks of all types.

> **Caution**: The XML security checks apply only to content that is sent with an HTTP content-type header of text/xml. If the content-type header is missing, or is set to a different value, all XML security checks are bypassed. If you plan to protect XML or Web 2.0 web applications, the webmasters of each web server that hosts those applications should ensure that the proper HTTP content-type header is sent.

XML format check

September 21, 2020

The XML Format check examines the XML format of incoming requests and blocks those requests that are not well formed or that do not meet the criteria in the XML specification for properly-formed XML documents. Some of those criteria are:

- An XML document must contain only properly-encoded Unicode characters that match the Unicode specification.
- No special XML syntax characters—such as < , > and &—can be included in the document except when used in XML markup.
- All begin, end, and empty-element tags must be correctly nested, with none missing or overlapping.
- XML element tags are case-sensitive. All beginning and end tags must match exactly.
- A single root element must contain all the other elements in the XML document.

A document that does not meet the criteria for well-formed XML does not meet the definition of an XML document. Strictly speaking, it is not XML. However, not all XML applications and web services enforce the XML well-formed standard, and not all handle poorly-formed or invalid XML correctly. Inappropriate handling of a poorly-formed XML document can cause security breaches. The purpose of the XML Format check is to prevent a malicious user from using a poorly-formed XML request to breach security on your XML application or web service.

If you use the wizard or the GUI, in the Modify XML Format Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Format Check:

- `set appfw profile <name> -xmlFormatAction [block] [log] [stats] [none]`

You cannot configure exceptions to the XML Format check. You can only enable or disable it.

XML denial-of-service check

September 21, 2020

The XML Denial of Service (XML DoS or XDoS) check examines incoming XML requests to determine whether they match the characteristics of a denial-of-service (DoS) attack, and blocks those requests that do. The purpose of the XML DoS check is to prevent an attacker from using XML requests to launch a denial-of-service attack on your web server or web site.

If you use the wizard or the GUI, in the Modify XML Denial-of-Service Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Denial-of-Service check:

- `set appfw profile <name> -xmlDoSAction [block] [log] [learn] [stats] [none]`

To configure individual XML Denial-of-Service rules, you must use the GUI. On the Checks tab of the Modify XML Denial-of-Service Check dialog box, select a rule and click Open to open the Modify XML Denial-of-Service dialog box for that rule. The individual dialog boxes differ for the different rules but are extremely simple. Some only allow you to enable or disable the rule; others allow you to modify a number by typing a new value in a text box.

The individual XML denial-of-service rules are:

- Maximum Element Depth

  Restrict the maximum number of nested levels in each individual element to 256. If this rule is enabled, and the App Firewall detects an XML request with an element that has more than the maximum number of allowed levels, it blocks the request. You can modify the maximum number of levels to any value from one (1) to 65,535.

- Maximum Element Name Length

  Restrict the maximum length of each element name to 128 characters. This includes the name within the expanded namespace, which includes the XML path and element name in the following format:

```
    {
 http://prefix.example.com/path/ }
 target_page.xml
<!--NeedCopy-->
```

```
The user can modify the maximum name length to any value between one
    (1) character and 65,535.
```

- Maximum # Elements

  Restrict the maximum number of any one type of element per XML document to 65,535. You can modify the maximum number of elements to any value between one (1) and 65,535.

- Maximum # Element Children

  Restrict the maximum number of children (including other elements, character information, and comments) each individual element is allowed to have to 65,535. You can modify the maximum number of element children to any value between one (1) and 65,535.

- Maximum # Attributes

  Restrict the maximum number of attributes each individual element is allowed to have to 256. You can modify the maximum number of attributes to any value between one (1) and 256.

- Maximum Attribute Name Length

  Restrict the maximum length of each attribute name to 128 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

- Maximum Attribute Value Length

  Restrict the maximum length of each attribute value to 2048 characters. You can modify the maximum attribute name length to any value between one (1) and 2,048.

- Maximum Character Data Length

  Restrict the maximum character data length for each element to 65,535. You can modify the length to any value between one (1) and 65,535.

- Maximum File Size

  Restrict the size of each file to 20 MB. You can modify the maximum file size to any value.

- Minimum File Size

  Require that each file be at least 9 bytes in length. You can modify the minimum file size to any positive integer representing a number of bytes.

- Maximum # Entity Expansions

  Limit the number of entity expansions allowed to the specified number. Default: 1024.

- Maximum Entity Expansion Depth

  Restrict the maximum number of nested entity expansions to no more than the specified number. Default: 32.

- Maximum # Namespaces

  Limit the number of namespace declarations in an XML document to no more than the specified number. Default: 16.

- Maximum Namespace URI Length

  Limit the URL length of each namespace declaration to no more than the specified number of characters. Default: 256.

- Block Processing Instructions

  Block any special processing instructions included in the request. This rule has no user-modifiable values.

- Block DTD

  Block any document type definitions (DTD) included with the request. This rule has no user-modifiable values.

- Block External Entities

  Block all references to external entities in the request. This rule has no user-modifiable values.

- SOAP Array Check

  Enable or disable the following SOAP array checks:

  - **Maximum SOAP Array Size.** The maximum total size of all SOAP arrays in an XML request before the connection is blocked. You can modify this value. Default: 20000000.
  - **Maximum SOAP Array Rank.** The maximum rank or dimensions of any single SOAP array in an XML request before the connection is blocked. You can modify this value. Default: 16.

XML cross-site scripting check

September 21, 2020

The XML Cross-Site Scripting check examines the user requests for possible cross-site scripting attacks in the XML payload. If it finds a possible cross-site scripting attack, it blocks the request.

To prevent misuse of the scripts on your protected web services to breach security on your web services, the XML Cross-Site Scripting check blocks scripts that violate the same origin rule, which states that scripts should not access or modify content on any server but the server on which they are located. Any script that violates the same origin rule is called a cross-site script, and the practice of using scripts to access or modify content on another server is called cross-site scripting. The reason cross-site scripting is a security issue is that a web server that allows cross-site scripting can be attacked with a script that is not on that web server, but on a different web server, such as one owned and controlled by the attacker.

The App Firewall offers various action options for implementing XML Cross-Site Scripting protection. You have the option to configure **Block**, **Log**, and **Stats** actions.

The App Firewall XML XSS check is performed on the payload of the incoming requests and attack strings are identified even if they are spread over multiple lines. The check looks for XSS attack strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

The CDATA section of the XML payload might be an attractive area of focus for the hackers because the scripts are not executable outside the CDATA section. A CDATA section is used for content that is to be treated entirely as character data. HTML mark up tag delimiters "<", ">", and "/>" will not cause the parser to interpret the code as HTML elements. The following example shows a CDATA Section with XSS attack string:

```
    <![CDATA[rn
    <script language="Javascript" type="text/javascript">alert ("Got you")</script>rn
    ]]>
<!--NeedCopy-->
```

Action Options

An action is applied when the XML Cross-Site Scripting check detects an XSS attack in the request. The following options are available for optimizing XML Cross-Site Scripting protection for your application:

- **Block**—Block action is triggered if the XSS tags are detected in the request.

- **Log**—Generate log messages indicating the actions taken by the XML Cross-Site Scripting check. If block is disabled, a separate log message is generated for each location (ELEMENT, ATTRIBUTE) in which the XSS violation is detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.
- **Stats**—Gather statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

Relaxation Rules

If your application requires you to bypass the Cross-Site Scripting check for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML Cross-Site Scripting check relaxation rules have the following parameters:

- **Name**—You can use literal strings or regular expressions to configure the name of the ELEMENT or the Attribute. The following expression exempts all ELEMENTS beginning with the string name_ followed by a string of uppercase or lowercase letters, or numbers, that is at least two and no more than fifteen characters long:

```
^name_[0-9A-Za-z]{ 2,15 } $
```

> **Note**
>
> The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:
>
> 1) XMLXSS: ABC IsRegex: NOTREGEX
> Location: ATTRIBUTE State: ENABLED
>
> 2) XMLXSS: ABC IsRegex: NOTREGEX
> Location: ELEMENT State: ENABLED
>
> 3) XMLXSS: abc IsRegex: NOTREGEX
> Location: ELEMENT State: ENABLED
>
> 4) XMLXSS: abc IsRegex: NOTREGEX
> Location: ATTRIBUTE State: ENABLED

- **Location**—You can specify the Location of the Cross-site Scripting Check exception in your XML payload. The option ELEMENT is selected by default. You can change it to ATTRIBUTE.
- **Comment**—This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> **Warning**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML Cross-Site Scripting check would otherwise have blocked.

Using the command line to configure the XML cross-site scripting check

To configure XML Cross-Site Scripting check actions and other parameters by using the command line

If you use the command-line interface, you can enter the following commands to configure the XML Cross-Site Scripting Check:

```
> set appfw profile <name> -XMLXSSAction (([block] [log] [stats])| [none])
```

To configure a XML Cross-Site Scripting check relaxation rule by using the command line

You can add relaxation rules to bypass inspection of XSS script attack inspection in a specific location. Use the bind or unbind command to add or delete the relaxation rule binding, as follows:

```
> bind appfw profile <name> -XMLXSS <string> [isRegex (REGEX | NOTREGEX)]
[-location ( ELEMENT | ATTRIBUTE )] -comment <string> [-state ( ENABLED |
DISABLED )]
```

```
> unbind appfw profile <name> -XMLXSS <String>
```

Example:

```
> bind appfw profile test_pr -XMLXSS ABC
```

After executing the above command, the following relaxation rule is configured. The rule is enabled, the name is treated as a literal (NOTREGEX), and ELEMENT is selected as the default location:

```
1)       XMLXSS:   ABC              IsRegex:   NOTREGEX

         Location:   ELEMENT        State:   ENABLED

`> unbind appfw profile test_pr -XMLXSS abc`

ERROR: No such XMLXSS check

`> unbind appfw profile test_pr -XMLXSS ABC`

 Done
<!--NeedCopy-->
```

Using the GUI to configure the XML cross-site scripting check

In the GUI, you can configure the XML Cross-Site scripting check in the pane for the profile associated with your application.

To configure or modify the XML Cross-Site Scripting check by using the GUI

1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a) If you just want to enable or disable **Block**, **Log**, and **Stats** actions for the **XML Cross-Site Scripting check**, you can select or clear check boxes in the table, click **OK**, and then click Save and Close to close the Security Check pane.

b) You can double click **XML Cross-Site Scripting**, or select the row and click **Action Settings**, to display the action options. After changing any of the action settings, click **OK** to save the changes and return to the Security Checks table.

You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save and Close** to close the Security Check pane.

To configure a XML Cross-Site Scripting relaxation rule by using the GUI

1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **XML Cross-Site Scripting** entry, or select it and click **Edit**.
4. In the **XML Cross-Site Scripting Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML Cross-Site Scripting relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML Cross-Site Scripting** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the Cross-Site Scripting patterns by using the GUI

You can use the GUI to view or customize the default list of XSS allowed attributes or allowed tags. You can also view or customize the default list of XSS denied Patterns.

The default lists are specified in **App Firewall** > **Signatures** > **Default Signatures**. If you do not bind any signature object to your profile, the default XSS Allowed and Denied list specified in the Default

Signatures object will be used by the profile for the Cross-Site Scripting security check processing. The Tags, Attributes, and Patterns, specified in the default signatures object, are read-only. You cannot edit or modify them. If you want to modify or change these, make a copy of the Default Signatures object to create a User-Defined signature object. Make changes in the Allowed or Denied lists in the new user-defined signature object and use this signature object in the profile that is processing the traffic for which you want to use these customized allowed and denied lists.

For more information about signatures, see Signatures.

To view default XSS patterns:

1. Navigate to **App Firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

The **Manage SQL/XSS Paths** table shows following three rows pertaining to XSS :

```
            xss/allowed/attribute

            xss/allowed/tag

            xss/denied/pattern
<!--NeedCopy-->
```

Select a row and click **Manage Elements** to display the corresponding XSS Elements (Tag, Attribute, Pattern) used by the App Firewall **Cross-Site Scripting** check.

To customize XSS Elements: You can edit the user-defined signature object to customize the allowed Tag, allowed Attributes and denied Patterns. You can add new entries or remove the existing ones.

1. **Navigate to App Firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.
2. Select the target XSS row.

a) Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding XSS element.

b) Click **Remove** to remove the selected row.

> **Warning**
>
> Be very careful when you remove or modify any default XSS element, or delete the XSS path to remove the entire row. The signatures, HTML Cross-Site Scripting security check, and XML Cross-Site Scripting security check rely on these Elements for detecting attacks to protect your applications. Customizing the XSS Elements can make your application vulnerable to Cross-Site Scripting attacks if the required pattern is removed during editing.

Using the log feature with the XML cross-site scripting check

When the log action is enabled, the XML Cross-Site Scripting security check violations are logged in the audit log as **APPFW_XML_XSS** violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

```
> Shell

> **tail -f /var/log/ns.log | grep APPFW_XML_XSS**
<!--NeedCopy-->
```

Example of a XML Cross-Site Scripting security check violation log message in Native log format showing <`blocked`> action

```
Oct  7 01:44:34 <local0.warn> 10.217.31.98 10/07/2015:01:44:34 GMT ns
    0-PPE-1 : default APPFW APPFW_XML_XSS 1154 0 :  10.217.253.69 3466-
    PPE1 - owa_profile http://10.217.31.101/FFC/login.html Cross-site
    script check failed for field script="Bad tag: script" <blocked>
<!--NeedCopy-->
```

Example of a XML Cross-Site Scripting security check violation log message in CEF log format showing <`not blocked`> action

```
Oct  7 01:46:52 <local0.warn> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|APPFW_XML_XSS|4|src=10.217.30.17 geolocation=Unknown spt
    =33141 method=GET request=http://10.217.31.101/FFC/login.html msg=
    Cross-site script check failed for field script="Bad tag: script"
    cn1=1607 cn2=3538 cs1=owa_profile cs2=PPE0 cs4=ERROR cs5=2015 act=**
    not blocked**
<!--NeedCopy-->
```

To access the log messages by using the GUI

The NetScaler GUI includes a useful tool (**Syslog Viewer**) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to the **App Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML Cross-Site Scripting** row and click **Logs**. When you access the logs directly from the XML Cross-Site Scripting check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.

- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click the Syslog messages link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **App Firewall** > **policies** > **Auditing**. In the **Audit Messages** section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML Cross-Site Scripting** check, filter by selecting **APPFW** in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_XSS** check box and click the **Apply** button, only log messages pertaining to the XML Cross-Site Scripting security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, **Client IP** etc. appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML cross-site scripting violations

When the stats action is enabled, the counter for the XML Cross-Site Scripting check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three XML Cross-Site Scripting violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML Cross-Site Scripting check statistics by using the command line

At the command prompt, type:

```
> **sh appfw stats**
```

To display stats for a specific profile, use the following command:

```
> **stat appfw profile** <profile name>
```

To display XML Cross-Site Scripting statistics by using the GUI

1. Navigate to **System** > **Security** > **App Firewall**.
2. In the right pane, access the **Statistics** Link.

3. Use the scroll bar to view the statistics about XML Cross-Site Scripting violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

XML SQL injection check

September 21, 2020

The XML SQL injection check examines the user requests for possible XML SQL Injection attacks. If it finds injected SQL in XML payloads, it blocks the requests.

A XML SQL attack can inject source code into a web application such that it can be interpreted and executed as a valid SQL query to perform a database operation with malicious intent. For example, XML SQL attacks can be launched to gain unauthorized access to the contents of a database or to manipulate the stored data. XML SQL Injection attacks are not only common, but can also be very harmful and costly.

Compartmentalizing the privileges of the database users can assist in protecting the database to some extent. All database users should be given only the required privileges to complete their intended tasks, so that they cannot execute SQL queries to perform other tasks. For example, a read-only user should not be allowed to write or manipulate data tables. The App Firewall XML SQL Injection check inspects all XML requests to provide special defenses against injection of unauthorized SQL code that might break security. If the App Firewall detects unauthorized SQL code in any XML request of any user, it can block the request.

The NetScaler App App Firewall inspects the presence of SQL keywords and special characters to identify the XML SQL Injection attack. A default set of keywords and special characters provides known keywords and special characters that are commonly used to launch XML SQL attacks. The App Firewall considers three characters, single straight quote (‘), backslash (), and semicolon (;) as special characters for SQL security check processing. You can add new patterns, and you can edit the default set to customize the XML SQL check inspection.

The App Firewall offers various action options for implementing XML SQL Injection protection. You can **Block** the request, **Log** a message in the ns.log file with details regarding the observed violations, and collect **Stats** to keep track of the number of observed attacks.

In addition to actions, there are several parameters that can be configured for XML SQL injection processing. You can check for **SQL wildcard characters**. You can change the XML SQL Injection type and select one of the 4 options (**SQLKeyword**, **SQLSplChar**, **SQLSplCharANDKeyword**, **SQLSplCharORKeyword**) to indicate how to evaluate the SQL keywords and SQL special characters when processing the XML payload. The XML **SQL Comments Handling** parameter gives you an option to specify the type of comments that need to be inspected or exempted during XML SQL Injection detection.

You can deploy relaxations to avoid false positives. The App Firewall XML SQL check is performed on the payload of the incoming requests, and attack strings are identified even if they are spread over multiple lines. The check looks for SQL Injection strings in the **element** and the **attribute** values. You can apply relaxations to bypass security check inspection under specified conditions. The logs and statistics can help you identify needed relaxations.

Action options

An action is applied when the XML SQL Injection check detects an SQL Injection attack string in the request. The following actions are available for configuring an optimized XML SQL Injection protection for your application:

Block—If you enable block, the block action is triggered only if the input matches the XML SQL injection type specification. For example, if **SQLSplCharANDKeyword** is configured as the XML SQL injection type, a request is not blocked if it does not contain any key words, even if SQL special characters are detected in the payload. Such a request is blocked if the XML SQL injection type is set to either **SQLSplChar**, or **SQLSplCharORKeyword**.

Log—If you enable the log feature, the XML SQL Injection check generates log messages indicating the actions that it takes. If block is disabled, a separate log message is generated for each location (**ELEMENT**, **ATTRIBUTE**) in which the XML SQL violation was detected. However, only one message is generated when the request is blocked. You can monitor the logs to determine whether responses to legitimate requests are getting blocked. A large increase in the number of log messages can indicate attempts to launch an attack.

Stats—If enabled, the stats feature gathers statistics about violations and logs. An unexpected surge in the stats counter might indicate that your application is under attack. If legitimate requests are getting blocked, you might have to revisit the configuration to see if you need to configure new relaxation rules or modify the existing ones.

XML SQL parameters

In addition to the block, log and stats actions, you can configure the following parameters for XML SQL Injection check:

Check for XML SQL wildcard Characters—Wild card characters can be used to broaden the selections of a structured query language (SQL-SELECT) statement. These wild card operators can be used in conjunction with **LIKE** and **NOT LIKE** operators to compare a value to similar values. The percent (%), and underscore (_) characters are frequently used as wild cards. The percent sign is analogous to the asterisk (*) wildcard character used with MS-DOS and to match zero, one, or multiple characters in a field. The underscore is similar to the MS-DOS question mark (?) wildcard character. It matches a single number or character in an expression.

For example, you can use the following query to do a string search to find all customers whose names contain the D character.

```
SELECT * from customer WHERE name like "%D%"
```

The following example combines the operators to find any salary values that have 0 as the second and third character.

```
SELECT * from customer WHERE salary like '_00%
```

Different DBMS vendors have extended the wildcard characters by adding extra operators. The NetScaler App App Firewall can protect against attacks that are launched by injecting these wildcard characters. The 5 default Wildcard characters are percent (%), underscore (_), caret (^), opening square bracket ([), and closing square bracket (]). This protection applies to both HTML and XML profiles.

The default wildcard chars are a list of literals specified in the ***Default Signatures**:

```
-   <wildchar type=” LITERAL” >%<wildchar>
-   <wildchar type=” LITERAL” >_<wildchar>
-   <wildchar type=” LITERAL” >^<wildchar>
-   <wildchar type=” LITERAL” >[<wildchar>
-   <wildchar type=” LITERAL” >]<wildchar>
<!--NeedCopy-->
```

Wildcard characters in an attack can be PCRE, like [^A-F]. The App Firewall also supports PCRE wildcards, but the literal wildcard chars above are sufficient to block most attacks.

> **Note**
>
> The XML SQL **wildcard character** check is different from the XML SQL **special character** check. This option must be used with caution to avoid false positives.

Check Request Containing SQL Injection Type—The App Firewall provides 4 options to implement the desired level of strictness for SQL Injection inspection, based on the individual need of the application. The request is checked against the injection type specification for detecting SQL violations. The 4 SQL injection type options are:

- **SQL Special Character and Keyword**—Both an SQL keyword and an SQL special character must be present in the inspected location to trigger SQL violation. This least restrictive setting is also the default setting.
- **SQL Special Character**—At least one of the special characters must be present in the processed payload string to trigger SQL violation.
- **SQL keyword**—At least one of the specified SQL keywords must be present in the processed payload string to trigger an SQL violation. Do not select this option without due consideration. To avoid false positives, make sure that none of the keywords are expected in the inputs.

- **SQL Special Character or Keyword**—Either the keyword or the special character string must be present in the payload to trigger the security check violation.

> **Tip**
>
> If you select the SQL Special Character option, the App Firewall skips strings that do not contain any special characters. Since most SQL servers do not process SQL commands that are not preceded by a special character, enabling this option can significantly reduce the load on the App Firewall and speed up processing without placing your protected web sites at risk.

SQL comments handling—By default, the App Firewall parses and checks all comments in XML data for injected SQL commands. Many SQL servers ignore anything in a comment, even if preceded by an SQL special character. For faster processing, if your XML SQL server ignores comments, you can configure the App Firewall to skip comments when examining requests for injected SQL. The XML SQL comments handling options are:

- **ANSI**—Skip ANSI-format SQL comments, which are normally used by UNIX-based SQL databases.
- **Nested**—Skip nested SQL comments, which are normally used by Microsoft SQL Server.
- **ANSI/Nested**—Skip comments that adhere to both the ANSI and nested SQL comment standards. Comments that match only the ANSI standard, or only the nested standard, are still checked for injected SQL.
- **Check all Comments**—Check the entire request for injected SQL, without skipping anything. This is the default setting.

> **Tip**
>
> In most cases, you should not choose the Nested or the ANSI/Nested option unless your back-end database runs on Microsoft SQL Server. Most other types of SQL server software do not recognize nested comments. If nested comments appear in a request directed to another type of SQL server, they might indicate an attempt to breach security on that server.

Relaxation rules

If your application requires you to bypass the XML SQL Injection inspection for a specific ELEMENT or ATTRIBUTE in the XML payload, you can configure a relaxation rule. The XML SQL Injection inspection relaxation rules have the following parameters:

- **Name:** You can use literal strings or regular expressions to configure the name of the **ELEMENT** or the **ATTRIBUTE**. The following expression exempts all **ELEMENTS** beginning with the string **PurchaseOrder_** followed by a string of numbers that is at least two and no more than ten characters long:

Comment: "Exempt XML SQL Check for Purchase Order Elements"

```
    XMLSQLInjection:  "PurchaseOrder_[0-9A-Za-z]{
 2,10 }
 "

    IsRegex:  REGEX          Location:  ELEMENT

    State:  ENABLED
<!--NeedCopy-->
```

Note: The names are case sensitive. Duplicate entries are not allowed, but you can use capitalization of the names and differences in location to create similar entries. For example, each of the following relaxation rules is unique:

```
1)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

2)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ELEMENT       State:  ENABLED

3)      XMLSQLInjection:  xyz    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED

4)      XMLSQLInjection:  XYZ    IsRegex:  NOTREGEX

        Location:  ATTRIBUTE     State:  ENABLED
<!--NeedCopy-->
```

- **Location**: You can specify the Location of the XML SQL Inspection exception in your XML payload. The option **ELEMENT** is selected by default. You can change it to **ATTRIBUTE**.
- **Comment**: This is an optional field. You can use up to a 255 character string to describe the purpose of this relaxation Rule.

> **Warning**
>
> Regular expressions are powerful. Especially if you are not thoroughly familiar with PCRE-format regular expressions, double-check any regular expressions you write. Make sure that they define exactly the name that you want to add as an exception, and nothing else. Careless use of Regular Expressions can have results that you do not want, such as blocking access to web content that you did not intend to block or allowing an attack that the XML SQL Injection inspection would otherwise have blocked.

Using the command line to configure the XML SQL Injection Check

To configure XML SQL Injection actions and other parameters by using the command line:

In the command line interface, you can use either the **set appfw profile** command or the **add appfw profile** command to configure the XML SQL Injection protections. You can enable the block, log, and stats action(s). Select the type of SQL attack pattern (key words, wildcard characters, special strings) you want to detect in the payloads. Use the **unset appfw profile** command to revert the configured settings back to their defaults. Each of the following commands sets only one parameter, but you can include multiple parameters in a single command:

- `set appfw profile <name> **-XMLSQLInjectionAction** (([block] [log] [stats])| [none])`
- `set appfw profile <name> -XMLSQLInjectionCheckSQLWildChars (ON |OFF)`
- `set appfw profile <name> -XMLSQLInjectionType ([SQLKeyword] | [SQLSplChar] | [SQLSplCharANDKeyword] | [SQLSplCharORKeyword])`
- `set appfw profile <name> -XMLSQLInjectionParseComments ([checkall] | [ansi|nested] | [ansinested])`

To configure a SQL Injection relaxation rule by using the command line

Use the bind or unbind command to add or delete relaxation rules, as follows:

```
-  bind appfw profile <name> -XMLSQLInjection <string> [isRegex (REGEX
   | NOTREGEX)] [-location ( ELEMENT | ATTRIBUTE )]  - comment <string>
   [-state ( ENABLED | DISABLED )]
-  unbind appfw profile <name> -XMLSQLInjection <String>
<!--NeedCopy-->
```

Example:

```
> bind appfw profile test_profile -XMLSQLInjection "PurchaseOrder_[0-9A
   -Za-z]{
 2,15 }
 " -isregex REGEX -location ATTRIBUTE

> unbind appfw profile test_profile  - XMLSQLInjection "PurchaseOrder_
   [0-9A-Za-z]{
 2,15 }
 " -location ATTRIBUTE
<!--NeedCopy-->
```

Using the GUI to configure the XMLSQL injection security check

In the GUI, you can configure the XML SQL Injection security check in the pane for the profile associated with your application.

To configure or modify the XML SQL Injection check by using the GUI

1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the Advanced Settings pane, click **Security Checks**.

The security check table displays the currently configured action settings for all the security checks. You have 2 options for configuration:

a. If you just want to enable or disable Block, Log, and Stats actions for XML SQL Injection, you can select or clear check boxes in the table, click OK, and then click Save and Close to close the Security Check pane.

b. If you want to configure additional options for this security check, double click XML SQL Injection, or select the row and click **Action Settings**, to display the following options:

Check for SQL Wildcard Characters—Consider SQL Wildcard characters in the payload to be attack patterns.

Check Request Containing—Type of SQL injection (SQLKeyword, SQLSplChar, SQLSplCharANDKeyword, or SQLSplCharORKeyword) to check.

SQL Comments Handling—Type of comments (Check All Comments, ANSI, Nested, or ANSI/Nested) to check.

After changing any of the above settings, click **OK** to save the changes and return to the Security Checks table. You can proceed to configure other security checks if needed. Click **OK** to save all the changes you have made in the Security Checks section, and then click **Save** and **Close** to close the Security Check pane.

To configure a XML SQL Injection relaxation rule by using the GUI

1. Navigate to **App Firewall** > **Profiles**, highlight the target profile, and click **Edit**.
2. In the **Advanced Settings** pane, click **Relaxation Rules**.
3. In the Relaxation Rules table, double-click the **XML SQL Injection** entry, or select it and click **Edit**.
4. In the **XML SQL Injection Relaxation Rules** dialogue box, perform **Add**, **Edit**, **Delete**, **Enable**, or **Disable** operations for relaxation rules.

To manage XML SQL Injection relaxation rules by using the visualizer

For a consolidated view of all the relaxation rules, you can highlight the **XML SQL Injection** row in the Relaxation Rules table, and click **Visualizer**. The visualizer for deployed relaxations offers you the option to **Add** a new rule or **Edit** an existing one. You can also **Enable** or **Disable** a group of rules by selecting a node and clicking the corresponding buttons in the relaxation visualizer.

To view or customize the SQL Injection patterns by using the GUI:

You can use the GUI to view or customize the SQL patterns.

The default SQL patterns are specified in **App Firewall** > **Signatures** > ***Default Signatures**. If you do not bind any signature object to your profile, the default SQL patterns specified in the Default Signatures object will be used by the profile for XML SQL Injection security check processing. The rules and patterns in the Default Signatures object are read-only. You cannot edit or modify them. If you want to modify or change these patterns, create a user-defined signature object by making a copy of the Default Signatures object and changing the SQL patterns. Use the user-defined signature object in the profile that processes the traffic for which you want to use these customized SQL patterns.

For more information, see Signatures.

To view default SQL patterns:

a. Navigate to **App Firewall** > **Signatures**, select ***Default Signatures**, and click **Edit**. Then click **Manage SQL/XSS Patterns**.

The Manage SQL/XSS Paths table shows following four rows pertaining to SQL Injection:

```
Injection (not_alphanum, SQL)/ Keyword

Injection (not_alphanum, SQL)/ specialstring

Injection (not_alphanum, SQL)/ transformrules/transform

Injection (not_alphanum, SQL)/ wildchar
<!--NeedCopy-->
```

b. Select a row and click **Manage Elements** to display the corresponding SQL patterns (keywords, special strings, transformation rules or the wildcard characters) used by the App Firewall SQL injection check.

To customize SQL Patterns: You can edit a user-defined signature object to customize the SQL key words, special strings, and wildcard characters. You can add new entries or remove the existing ones. You can modify the transformation rules for the SQL special strings.

a. Navigate to **App Firewall** > **Signatures**, highlight the target user-defined signature, and click **Edit**. Click **Manage SQL/XSS Patterns** to display the **Manage SQL/XSS paths** table.

b. Select the target SQL row.

i. Click **Manage Elements**, to **Add**, **Edit** or **Remove** the corresponding SQL element.

ii. Click **Remove** to remove the selected row.

> **Warning**
>
> You must be very careful when removing or modifying any default SQL element, or deleting the SQL path to remove the entire row. The signature rules as well as the XML SQL Injection security check rely on these elements for detecting SQL Injection attacks to protect your applications. Customizing the SQL patterns can make your application vulnerable to XML SQL attacks if the required pattern is removed during editing.

Using the log feature with the XML SQL injection check

When the log action is enabled, the **XML SQL Injection** security check violations are logged in the audit log as **APPFW_XML_SQL** violations. The App Firewall supports both Native and CEF log formats. You can also send the logs to a remote syslog server.

To access the log messages by using the command line:

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the XML Cross-Site Scripting violations:

```
> Shell

> tail -f /var/log/ns.log | grep APPFW_XML_SQL
<!--NeedCopy-->
```

To access the log messages by using the GUI

The NetScaler GUI includes a useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- Navigate to **App Firewall** > **Profiles**, select the target profile, and click **Security Checks**. Highlight the **XML SQL Injection** row and click **Logs**. When you access the logs directly from the XML SQL Injection check of the profile, the GUI filters out the log messages and displays only the logs pertaining to these security check violations.
- You can also access the Syslog Viewer by navigating to **System** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the Syslog Viewer, which displays all log messages, including other security check violation logs. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **App Firewall** > **Policies** > **Auditing**. In the Audit Messages section, click the **Syslog messages** link to display the **Syslog Viewer**, which displays all log messages, including other security check violation logs.

The XML based Syslog Viewer provides various filter options for selecting only the log messages that are of interest to you. To select log messages for the **XML SQL Injection** check, filter by selecting

APPFW in the dropdown options for **Module**. The **Event Type** list offers a rich set of options to further refine your selection. For example, if you select the **APPFW_XML_SQL** check box and click the **Apply** button, only log messages pertaining to the **XML SQL Injection** security check violations appear in the Syslog Viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **Event Type**, **Event ID**, and **Client IP** appear below the log message. You can select any of these options to highlight the corresponding information in the log message.

Statistics for the XML SQL injection violations

When the stats action is enabled, the counter for the **XML SQL Injection** check is incremented when the App Firewall takes any action for this security check. The statistics are collected for Rate and Total count for Traffic, Violations, and Logs. The size of an increment of the log counter can vary depending on the configured settings. For example, if the block action is enabled, a request for a page that contains three **XML SQL Injection** violations increments the stats counter by one, because the page is blocked as soon as the first violation is detected. However, if block is disabled, processing the same request increments the statistics counter for violations and the logs by three, because each violation generates a separate log message.

To display XML SQL Injection check statistics by using the command line

At the command prompt, type:

```
> sh appfw stats
```

To display stats for a specific profile, use the following command:

```
> stat appfw profile <profile name>
```

To display XML SQL Injection statistics by using the GUI

1. Navigate to **System** > **Security** > **App Firewall**.
2. In the right pane, access the **Statistics** Link.
3. Use the scroll bar to view the statistics about **XML SQL Injection** violations and logs. The statistics table provides real-time data and is updated every 7 seconds.

XML attachment check

September 21, 2020

The XML Attachment check examines incoming requests for malicious attachments, and it blocks those requests that contain attachments that might breach applications security. The purpose of the

XML Attachment check is to prevent an attacker from using an XML attachment to breach security on your server.

If you use the wizard or the GUI, in the Modify XML Attachment Check dialog box, on the General tab you can enable or disable the Block, Learn, Log, Statistics, and Learn actions:

If you use the command-line interface, you can enter the following command to configure the XML Attachment Check:

- `set appfw profile <name> -xmlAttachmentAction [block] [learn] [log] [ stats] [none]`

You must configure the other XML Attachment check settings in the GUI. In the
`Modify XML Attachment` Check dialog box, on the
Checks tab, you can configure the following settings:

- **Maximum Attachment Size.** Allow attachments that are no larger than the maximum attachment size you specify. To enable this option, first select the Enabled check box, and then type the maximum attachment size in bytes in the `Size` text box.
- **Attachment Content Type.** Allow attachments of the specified content type. To enable this option, first select the Enabled check box, and then enter a regular expression that matches the Content-Type attribute of the attachments that you want to allow.
  - You can type the URL expression directly in the text window. If you do so, you can use the `Regex Tokens` menu to enter a number of useful regular expressions at the cursor instead of typing them manually.
  - You can click Regex Editor to open the `Add Regular Expression` dialog box and use it to construct the URL expression.

Web services interoperability check

September 21, 2020

The Web Services Interoperability (WS-I) check examines both requests and responses for adherence to the WS-I standard, and blocks those requests and responses that do not adhere to this standard. The purpose of the WS-I check is to block requests that might not interact with other XML appropriately. An attacker can use inconsistencies in interoperability to launch an attack on your XML application.

If you use the wizard or the GUI, in the Modify Web Services Interoperability Check dialog box, on the General tab you can enable or disable the Block, Log, Statistics, and Learn actions.

If you use the command-line interface, you can enter the following command to configure the Web Services Interoperability check:

- `set appfw profile <name> -xmlWSIAction [block] ][log] [learn] [stats] [ none]`

To configure individual Web Services Interoperability rules, you must use the GUI. On the Checks tab of the Modify Web Services Interoperability Check dialog box, select a rule and click Enable or Disable to enable or disable the rule. You can also click Open to open the Web Services Interoperability Detail message box for that rule. The message box displays read-only information about the rule. You cannot modify or make other configuration changes to any of these rules.

The WS-I check uses the rules listed in WS-I Basic Profile 1.0. WS-I delivers best practices for developing interoperable Web Services solutions. WS-I checks are performed only on SOAP Messages.

Description of each WSI standard rule is provided below:

Rule	Description
BP1201	Message body should be a soap:envelope with namespace.
R1000	When an ENVELOPE is a Fault, the soap:Fault element MUST NOT have element children other than faultcode, faultstring, faultactor and detail.
R1001	When an ENVELOPE is a Fault, the element children of the soap:Fault element MUST be unqualified.
R1003	A RECEIVER MUST accept fault messages that have any number of qualified or unqualified attributes, including zero, appearing on the detail element. The namespace of qualified attributes can be anything other than the namespace of the qualified document element Envelope.
R1004	When an ENVELOPE contains a faultcode element, the content of that element SHOULD be either one of the fault codes defined in SOAP 1.1 (supplying additional information if necessary in the detail element), or a Qname whose namespace is controlled by the fault's specifying authority (in that order of preference).

Rule	Description
R1005	An ENVELOPE MUST NOT contain soap:encodingStyle attribute on any of the elements whose namespace is the same as the namespace of the qualified document element Envelope.
R1006	An ENVELOPE MUST NOT contain soap:encodingStyle attributes on any element that is a child of soap:Body.
R1007	An ENVELOPE described in an rpc-literal binding MUST NOT contain soap:encodingStyle attribute on any element that is a grandchild of soap:Body.
R1011	An ENVELOPE MUST NOT have any element children of soap:Envelope following the soap:Body element.
R1012	A MESSAGE MUST be serialized as either UTF-8 or UTF-16.
R1013	An ENVELOPE containing a soap:mustUnderstand attribute MUST only use the lexical forms 0 and 1.
R1014	The children of the soap:Body element in an ENVELOPE MUST be namespace qualified.
R1015	A RECEIVER MUST generate a fault if they encounter an envelope whose document element is not soap:Envelope.
R1031	When an ENVELOPE contains a faultcode element the content of that element SHOULD NOT use of the SOAP 1.1 dot notation to refine the meaning of the fault.
R1032	The soap:Envelope, soap:Header, and soap:Body elements in an ENVELOPE MUST NOT have attributes in the same namespace as that of the qualified document element Envelope

Rule	Description
R1033	An ENVELOPE SHOULD NOT contain the namespace declaration: `xmlns:xml=http://www.w3.org/XML/1998/namespace.`
R1109	The value of the SOAPAction HTTP header field in a HTTP request MESSAGE MUST be a quoted string.
R1111	An INSTANCE SHOULD use a 200 OK HTTP status code on a response message that contains an envelope that is not a fault.
R1126	An INSTANCE MUST return a 500 Internal Server Error HTTP status code if the response envelope is a Fault.
R1132	A HTTP request MESSAGE MUST use the HTTP POST method.
R1140	A MESSAGE SHOULD be sent using HTTP/1.1.
R1141	A MESSAGE MUST be sent using either HTTP/1.1 or HTTP/1.0.
R2113	An ENVELOPE MUST NOT include the soapenc:arrayType attribute.
R2211	An ENVELOPE described with an rpc-literal binding MUST NOT have the xsi:nil attribute with a value of 1 or true on the part accessors.
R2714	For one-way operations, an INSTANCE MUST NOT return a HTTP response that contains an envelope. Specifically, the HTTP response entity-body must be empty.
R2729	An ENVELOPE described with an rpc-literal binding that is a response MUST have a wrapper element whose name is the corresponding wsdl:operation name suffixed with the stringResponse.
R2735	An ENVELOPE described with an rpc-literal binding MUST place the part accessor elements for parameters and return value in no namespace.

Rule	Description
R2738	An ENVELOPE MUST include all soapbind:headers specified on a wsdl:input or wsdl:output of a wsdl:operation of a wsdl:binding that describes it.
R2740	A wsdl:binding in a DESCRIPTION SHOULD contain a soapbind:fault describing each known fault.
R2744	A HTTP request MESSAGE MUST contain a SOAPAction HTTP header field with a quoted value equal to the value of the soapAction attribute of soapbind:operation, if present in the corresponding WSDL description.

XML message validation check

September 21, 2020

The XML Message Validation check examines requests that contain XML messages to ensure that they are valid. If a request contains an invalid XML message, the App Firewall blocks the request. The purpose of the XML Validation check is to prevent an attacker from using specially constructed invalid XML messages to breach the security of your application.

If you use the wizard or the GUI, in the Modify XML Message Validation Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions.

If you use the command-line interface, you can enter the following command to configure the XML Message Validation Check:

- `set appfw profile <name> -xmlValidationAction [block] [log] [stats] [ none]`

You must use the GUI to configure the other XML Validation check settings. In the Modify XML Message Validation Check dialog box, on the Checks tab, you can configure the following settings:

- **XML Message Validation.** Use one of the following options to validate the XML message:
  - **SOAP Envelope.** Validate only the SOAP envelope of XML messages.
  - **WSDL.** Validate XML messages by using an XML SOAP WSDL. If you choose WSDL validation, in the WSDL Object drop-down list you must choose a WSDL. If you want to validate against

a WSDL that has not already been imported to the App Firewall, you can click the Import button to open the Manage WSDL Imports dialog box and import your WSDL. See "WSDL topic" for more information.

    - If you want to validate the entire URL, leave the Absolute radio button in the End Point Check button array selected. If you want to validate only the portion of the URL after the host, select the Relative radio button.
    - If you want the App Firewall to enforce the WSDL strictly, and not allow any additional XML headers not defined in the WSDL, you must clear the Allow additional headers not defined in the WSDL check box.
      Caution: If you uncheck the
      Allow Additional Headers not defined in the WSDL check box, and your WSDL does not define all XML headers that your protected XML application or Web 2.0 application expects or that a client sends, you may block legitimate access to your protected service.
  - **XML Schema.** Validate XML messages by using an XML schema. If you choose XML schema validation, in the XML Schema Object drop-down list you must choose an XML schema. If you want to validate against an XML schema that has not already been imported to the App Firewall, you can click the Import button to open the Manage XML Schema Imports dialog box and import your WSDL. See WSDL for more information.
- **Response Validation.** By default, the App Firewall does not attempt to validate responses. If you want to validate responses from your protected application or Web 2.0 site, select the Validate Response check box. When you do, the Reuse the XML Schema specified in request validation check box and the XML Schema Object drop-down list are activated.
  - Check the Reuse XML Schema check box to use the schema you specified for request validation to do response validation as well.
    Note: If you check this check box, the
    XML Schema Object drop-down list is grayed out.
  - If you want to use a different XML schema for response validation, use the XML Schema Object drop-down list to select or upload that XML schema.

XML SOAP fault filtering check

September 21, 2020

The XML SOAP fault filtering check examines responses from your protected web services and filters out XML SOAP faults. This prevents leaking of sensitive information to attackers.

If you use the wizard or the GUI, in the Modify XML SOAP Fault Filtering Check dialog box, on the General tab you can enable or disable the Block, Log, and Statistics actions, and the Remove action, which removes SOAP faults before forwarding the response to the user.

If you use the command-line interface, you can enter the following command to configure the XML SOAP Fault Filtering Check:

```
set appfw profile <name> -XMLSOAPFaultAction [block] [log] [stats] [none]
```

You cannot configure exceptions to the XML SOAP Fault Filtering check. You can only enable or disable it.

Managing content types

September 21, 2020

Web servers usually add a Content-Type header that contains a MIME/type definition for the type of content in each file that the web server serves to users. Web servers serve many different types of content. For example, standard HTML is assigned the "text/html" MIME type. JPG images are assigned the "image/jpeg" or "image/jpg" content type. A normal web server can serve dozens or hundreds of different types of content, all defined in the Content Type header by an assigned MIME/type.

Many App Firewall filtering rules are designed to filter specific types of content. Because filtering rules that apply to one type of content (such as HTML) are often inappropriate when filtering a different type of content (such as images), the App Firewall attempts to determine the content type of requests and responses before it filters them. When a web server or browser does not add a Content-Type header to a request or response, the App Firewall applies a default content type to the connection and filters the content accordingly.

The default content type is normally "application/octet-stream", the most generic MIME/type definition.This MIME/type is appropriate for any type of content that a web server is likely to serve, but also does not provide much information to the App Firewall to allow it to choose appropriate filtering. If a protected web server on your network is configure to add accurate content type headers to the content it serves, or serves only one type of content, you can create a profile for that web server and assign a different default content type to it to improve both the speed and the accuracy of filtering.

You can also configure a list of allowed request content types for a specific profile. When this feature is configured, if the App Firewall filters a request that does not match one of the allowed content types, it blocks the request. After upgrade from release 10.5 to 11.0, unknown content-types which are not in the default allowed content-type list do not bind. You can add other content-types which you want to be allowed to the relaxed rules.

Requests must always be of either the "application/x-www-form-urlencoded", "multipart/form-data", or "text/x-gwt-rpc" types. The App Firewall blocks any request that has any other content type designated.

> **Note**
>
> You cannot include the "application/x-www-form-urlencoded" or "multipart/form-data" content types on the allowed response content types list.

To set the default request content type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -requestContentType <type>`
- `save ns config`

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

To remove the user-defined default request content type by using the command line interface

At the command prompt, type the following commands:

- `unset appfw profile <name> -requestContentType <type>`
- `save ns config`

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -requestContentType "text/html"
save ns config
<!--NeedCopy-->
```

> **Note**
>
> Always use last content-type header for processing and remove remaining content-type headers if any that ensures that the backend server receives a request with only one content-type.
>
> To block requests that can be bypassed, add an App Firewall policy with rule as HTTP.REQ.HEADER

> ("content-type").COUNT.GT(1)' and profile as *appfw_block*.
>
> If a request is received without a Content-Type header or if the request has Content-Type header without any value, App Firewall applies the configured **RequestContentType** value and processes the request accordingly.

To set the default response content type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -responseContentType <type>`
- `save ns config`

Example

The following example sets the "text/html" content type as the default for the specified profile:

```
set appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

To remove the user-defined default response content type by using the command line interface

At the command prompt, type the following commands:

- `unset appfw profile <name> -responseContentType <type>`
- `save ns config`

Example

The following example unsets the default content type of "text/html" for the specified profile, allowing the type to revert to "application/octet-stream":

```
unset appfw profile profile1 -responseContentType "text/html"
save ns config
<!--NeedCopy-->
```

To add a content type to the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- `bind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

Example

The following example adds the "text/shtml" content type to the allowed content types list for the specified profile:

```
bind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

To remove a content type from the allowed content types list by using the command line interface

At the command prompt, type the following commands:

- `unbind appfw profile <name> -ContentType <contentTypeName>`
- `save ns config`

Example

The following example removes the "text/shtml" content type from the allowed content types list for the specified profile:

```
unbind appfw profile profile1 -contentType "text/shtml"
save ns config
<!--NeedCopy-->
```

To manage the default and allowed content types by using the GUI

1. Navigate to **Security** > **App Firewall** > **Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**. The **Configure App Firewall Profile** dialog box is displayed.
3. The **Configure App Firewall Profile** dialog box, click the **Settings** tab.
4. On the **Settings** tab, scroll down about halfway to the Content Type area.
5. In the Content Type area, configure the default request or response content type:
   - To configure the default request content type, type the MIME/type definition of the content type you want to use in the Default Request text box.
   - To configure the default response content type, type the MIME/type definition of the content type you want to use in the Default Response text box.

    - To create a new allowed content type, click **Add**. The **Add Allowed Content Type** dialog box is displayed.
    - To edit an existing allowed content type, select that content type, and then click **Open**. The **Modify Allowed Content Type** dialog box is displayed.
6. To manage the allowed content types, click Manage Allowed Content Types.
7. To add a new content type or modify an existing content type, click Add or Open, and in the **Add Allowed Content Type** or **Modify Allowed Content Type** dialog box, do the following steps.
    a) Select/clear the Enabled check box to include the content type in, or exclude it from, the list of allowed content types.
    b) In the Content Type text box, type a regular expression that describes the content type that you want to add, or change the existing content type regular expression.
    Content types are formatted exactly as MIME type descriptions are.

    > **Note**: You can include any valid MIME type on the allowed contents type list. Since many types of document can contain active content and therefore could potentially contain malicious content, you should exercise caution when adding MIME types to this list.

    c) In the Comments text box, add an optional comment that describes the reason for adding this particular MIME type to the allowed contents type list.
    d) Click **Create** or **OK** to save your changes.
8. Click **Close** to close the Manage Allowed Content Types dialog box and return to the **Settings** tab.
9. Click **OK** to save your changes.

Profiles

September 21, 2020

A profile is a collection of security settings that are used to protect specific types of web content or specific parts of your web site. In a profile, you determine how the App Firewall applies each of its filters (or checks) to requests to your web sites, and responses from them. The App Firewall supports two types of profile: four built-in (default) profiles that do not require further configuration, and user-defined profiles that do require further configuration.

Built-in profiles

The four App Firewall built-in profiles provide simple protection for applications and web sites that either do not require protection, or that should not be directly accessed by users at all. These profile types are:

- **APPFW_BYPASS.** Skips all App Firewall filtering and sends the unmodified traffic to the protected application or web site, or to the client.
- **APPFW_RESET.** Resets the connection, requiring that the client re-establish his or her session by visiting a designated start page.
- **APPFW_DROP.** Drops all traffic to or from the protected application or web site, and sends no response of any kind to the client.
- **APPFW_BLOCK.** Blocks traffic to or from the protected application or web site.

You use the built-in profiles exactly as you do user-defined profiles, by configuring a policy that selects the traffic to which you want to apply the profile and then associating the profile with your policy. Since you do not have to configure a built-in policy, it provides a quick way to allow or block specified types of traffic or traffic that is sent to specific applications or web sites.

User-defined profiles

User-defined profiles are profiles that are build and configured by users. Unlike the default profiles, you must configure a user-defined profile before it will be of use filtering traffic to and from your protected applications.

There are three types of user-defined profile:

- **HTML.** Protects HTML-based web pages.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0.** Protects Web 2.0 content that combines HTML and XML content, such as ATOM feeds, blogs, and RSS feeds.

The App Firewall has a number of security checks, all of which can be enabled or disabled, and configured in a number of ways in each profile. Each profile also has a number of settings that control how it handles different types of content. Finally, rather than manually configuring all of the security checks, you can enable and configure the learning feature. This feature observes normal traffic to your protected web sites for a period of time, and uses those observations to provide you with a tailored list of recommended exceptions (*relaxations*) to some security checks, and additional rules for other security checks.

During initial configuration, whether by using the App Firewall Wizard or manually, you normally create one general purpose profile to protect all content on your web sites that is not covered by a more specific profile. After that, you can create as many specific profiles as you want to protect more specialized content.

The Profiles pane consists of a table that contains the following elements:

Name. Displays all the App Firewall profiles configured in the appliance.

Bound signature. Displays the signatures object that is bound to the profile in the previous column, if any.

Policies. Displays the App Firewall policy that invokes the profile in the leftmost column of that row, if any.

Comments. Displays the comment associated with the profile in the leftmost column of that row, if any.

Profile Type. Displays the type of profile. Types are Built-In, HTML, XML, and Web 2.0.

Above the table is a row of buttons and a drop-down list that allow you to create, configure, delete, and view information about your profiles:

- **Add.** Add a new profile to the list.
- **Edit.** Edit the selected profile.
- **Delete.** Delete the selected profile from the list.
- **Statistics.** View the statistics for the selected profile.
- **Action.** Drop-down list that contains additional commands. Currently allows you to import a profile that was exported from another App Firewall configuration.

Creating App Firewall profiles

September 22, 2020

You can create an App Firewall profile in one of two ways: by using the command line, and by using the GUI. Creating a profile by using the command line requires that you specify options on the command line. The process is similar to that of configuring an existing profile, and with a few exceptions the two commands take the same parameters.

Creating a profile by using the GUI requires that you specify only two options. You specify basic or advanced *defaults*, the default configuration for the various security checks and settings that are part of a profile, and choose the profile *type* to match the type of content that the profile is intended to protect. You can also, optionally, add a comment. After you create the profile, you must then configure it by selecting it in the data pane, and then clicking Edit.

If you plan to use the learning feature or to enable and configure a large number of advanced protections, you should choose advanced defaults. In particular, if you plan to configure either of the SQL injection checks, either of the cross-site scripting checks, any check that provides protection against Web form attacks, or the cookie consistency check, you should plan to use the learning feature. Unless you include the proper exceptions for your protected Web sites when configuring these checks, they can block legitimate traffic. Anticipating all of the necessary exceptions without creating any that are too broad is difficult. The learning feature makes this task much easier. Otherwise, basic defaults are quick and should provide the protection that your web applications need.

There are three profile types:

- **HTML.** Protects standard HTML-based web sites.
- **XML.** Protects XML-based web services and web sites.
- **Web 2.0 (HTML XML).** Protects sites that contain both HTML and XML elements, such as ATOM feeds, blogs, and RSS feeds.

There are also a few restrictions on the name that you can give to a profile. A profile name cannot be the same as the name assigned to any other profile or action in any feature on the NetScaler appliance. Certain action or profile names are assigned to built-in actions or profiles, and can never be used for user profiles. A complete list of disallowed names can be found in the App Firewall ProfileSupplemental Information. If you attempt to create a profile with a name that has already been used for an action or a profile, an error message is displayed and the profile is not created.

To create an App Firewall profile by using the command line interface

At the command prompt, type the following commands:

- `add appfw profile <name> [-defaults ( basic | advanced )]`
- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `set appfw profile <name> -comment "<comment>"`
- `save ns config`

Example

The following example adds a profile named pr-basic, with basic defaults, and assigns a profile type of HTML. This is the appropriate initial configuration for a profile to protect an HTML Web site.

```
add appfw profile pr-basic -defaults basic -comment "Simple profile for
    web sites."
set appfw profile pr-basic -type HTML
save ns config
<!--NeedCopy-->
```

To create an App Firewall profile by using the GUI

Creating an App Firewall profile requires that you specify only a few configuration details.

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, click **Add**.
3. In the **Create App Firewall Profile** dialog box, type a name for your profile.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 31 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Choose the profile type from the drop-down list.
5. Click **Create**, and then click **Close**.

Configuring App Firewall profiles

September 22, 2020

To configure a user-defined App Firewall profile, first configure the security checks, which are called *deep protections* or *advanced protections* in the App Firewall wizard. Certain checks require configuration if you are to use them at all. Others have default configurations that are safe but limited in scope; your web sites might need or benefit from a different configuration that takes advantage of additional features of certain security checks.

After you have configured the security checks, you can also configure a number of other settings that control the behavior, not of a single security check, but the App Firewall feature. The default configuration is sufficient to protect most web sites, but you should review them to make sure that they are right for your protected web sites.

For more information about the App Firewall security checks, see "Advanced Protections."

To configure an App Firewall profile by using the command line

At the command prompt, type the following commands:

- `set appfw profile <name> <arg1> [<arg2> ...]`

  where:

  - `<arg1>` = a parameter and any associated options.
  - `<arg2>` = a second parameter and any associated options.
  - … = additional parameters and options.

  For descriptions of the parameters to use when configuring specific security checks, see "Advanced Protections."

- `save ns config`

Example

The following example shows how to enable blocking for the HTML SQL Injection and HTML Cross-Site Scripting checks in a profile named
pr-basic. This command enables blocking for those actions while making no other changes to the profile.

```
set appfw profile pr-basic -crossSiteScriptingAction block -
    SQLInjectionAction block
<!--NeedCopy-->
```

To configure an App Firewall profile by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Profiles**.
2. In the details pane, select the profile that you want to configure, and then click **Edit**.
3. In the **Configure App Firewall Profile** dialog box, on the **Security Checks** tab, configure the security checks.
    - To enable or disable an action for a check, in the list, select or clear the check box for that action.
    - To configure other parameters for those checks that have them, in the list, click the blue chevron to the far right of that check. In the dialog box that appears, configure the parameters. These vary from check to check.
      You can also select a check and, at the bottom of the dialog box, click Open to display the **Configure Relaxation** dialog box or **Configure Rule** dialog box for that check. These dialog boxes also vary from check to check. Most of them include a **Checks** tab and a **General** tab. If the check supports relaxations or user-defined rules, the **Checks** tab includes an **Add** button, which opens yet another dialog box, in which you can specify a relaxation or rule for the check. (A relaxation is a rule for exempting specified traffic from the check.) If relaxations have already been configured, you can select one and click **Open** to modify it.
    - To review learned exceptions or rules for a check, select the check, and then click **Learned Violations**. In the **Manage Learned Rules** dialog box, select each learned exception or rule in turn.
        - To edit the exception or rule, and then add it to the list, click **Edit & Deploy**.
        - To accept the exception or rule without modification, click **Deploy**.
        - To remove the exception or rule from the list, click **Skip**.
    - To refresh the list of exceptions or rules to be reviewed, click **Refresh**.
    - open the Learning Visualizer and use it to review learned rules, click **Visualizer**.
    - review the log entries for connections that matched a check, select the check, and then click Logs. You can use this information to determine which checks are matching attacks, so that you can enable blocking for those checks. You can also use this information to determine which checks are matching legitimate traffic, so that you can configure an appropriate exemption to allow those legitimate connections. For more information about the logs, see "Logs, Statistics, and Reports."
    - To completely disable a check, in the list, clear all of the check boxes to the right of that check.

4. On the **Settings** tab, configure the profile settings.
   - To associate the profile with the set of signatures that you previously created and configured, under Common Settings, choose that set of signatures in the **Signatures** drop-down list.

     > **Note**: You may need to use the scroll bar on the right of the dialog box to scroll down to display the Common Settings section.

   - To configure an HTML or XML Error Object, select the object from the appropriate drop-down list.

     > **Note**: You must first upload the error object that you want to use in the Imports pane. For more information about importing error objects, see Imports

   - To configure the default XML Content Type, type the content type string directly into the Default Request and Default Response text boxes, or click Manage Allowed Content Types to manage the list of allowed content types. "»More…."
5. If you want to use the learning feature, click Learning, and configure the learning settings for the profile, as described in "Configuring and Using the Learning Feature".
6. Click **OK** to save your changes and return to the **Profiles** pane.

Changing an App Firewall profile type

September 22, 2020

If you chose the wrong profile type for an App Firewall profile, or the type of content on the protected web site has changed, you can change the profile type.

> **Note**: When you change the profile type, you lose all configuration settings and learned relaxations or rules for the features that the new profile type does not support. For example, if you change the profile type from Web 2.0 to XML, you lose any configuration options for Start URL, Form Field Consistency Check, and the other HTML-specific security checks. The configuration for any options that is supported by both the old and the new profile types remains unchanged.

To change an App Firewall profile type by using the command line interface

At the command prompt, type the following commands:

- `set appfw profile <name> -type ( HTML | XML | HTML XML )`
- `save ns config`

Example

The following example changes the type of a profile named pr-basic, from HTML to HTML XML, which is equivalent to the Web 2.0 type in the GUI.

```
set appfw profile pr-basic -type HTML XML
save ns config
<!--NeedCopy-->
```

To change an App Firewall profile type by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Policies**.
2. In the details pane, click **Action**, and then **Change Profile Type**.
3. In the **Change App Firewall Profile Type** dialog box, **Profile Type** drop-down list, select a new profile type.
4. Click **OK** to save your changes and return to the **Profiles** pane.

Exporting and importing an App Firewall profile

September 21, 2020

You can replicate the entire configuration of an App Firewall profile (including all bound objects, such as HTML error object, XML error object, WSDL or XML schema, signatures, and so on) across multiple appliances. You can select a target profile and export the configuration to save it in your computer's local file system, or you can transfer the archived configuration to store it on a server. Similarly, you can browse your computer's local file system or import the archive from the server to select a previously exported profile and import it into your NetScaler appliance.

The option to export the entire profile configuration and then import it into another appliance can be useful in various use cases. For example, you might want to configure an App Firewall profile in a test bed set-up to test and validate that it is working as expected. Once you are satisfied, you can export the profile and import the profile configuration to your production NetScaler appliances. This functionality is also useful for backing up your configuration. You can export the profile before making changes, so that you can easily roll back the configuration to a known state if necessary.

> **Note**
>
> App Firewall profiles that are exported and archived from one build cannot be restored to a system running a different build, because changes introduced in the newer releases can lead to compatibility issues. If you attempt to restore an archived profile to a different build than the one from which it was exported, an error message is logged in ns.log.

The export and import profile functionality is available in both the GUI (GUI) and the command line interface (CLI). The GUI is recommended, because it offers easy to use **Action** options. With a click of a button you can **Export** or **Import** the entire configuration of a profile.

Exporting App Firewall profiles with the CLI

If you use CLI to **export** a profile, you must **archive** the configuration and then **export** it. To **import** a profile, you must **import** the archive into the NetScaler appliance and then execute the **restore** command to extract the configuration. The following set of CLI commands can be used for exporting, importing and managing the profile configurations.

CLI commands to export archives:

- `archive appfw profile <name> <archivename> [-comment <string>]`
- `export appfw archive <name> <target>`

CLI commands to import archives:

- `import appfw archive <src> <name> [-comment <string>]`
- `restore appfw profile <archivename>`

CLI commands to manage archives:

- `show appfw archive`
- `rm appfw archive <name>`

Exporting a profile from one appliance and importing it to another requires five steps in CLI. The first 3 steps are performed on the source appliance on which profile configuration is originally created, and the next 2 steps are performed on the target appliance on which the profile configuration is to be replicated.

Export profile from the source NetScaler appliance:

Step 1: Create an archive of the configured profile.

Step 2: Export the archive to the NetScaler file system.

Step 3: Use a file transfer utility such as scp to transfer the exported archive file from NetScaler appliance A to the target NetScaler appliance.

Import profile to the target NetScaler appliance:

Step 4: Execute the import command to import the archived file. You can import the archive from your NetScaler's local file system, or you can use HTTP or HTTPS protocol to import the archive from a server by using the URL.

Step 5: Execute the restore command to restore the profile configuration from the imported archive

To export an App Firewall profile by using the command line interface:

First, **archive** the profile's configuration, and then **export** the archive to a target location. At the command prompt, type the following commands:

```
archive appfw profile <profileName> <archiveName>
```

where:

- `<profileName>` is the name of the profile to archive.
- `<archiveName>` is the name of the archive file to create.

Execution of the above command creates 2 instances of the archive file. One in /var/tmp folder and another in the /var/archive/appfw folder.

export appfw archive `<archiveName> <target>`

where:

- `<archiveName>` is the name of the archive to export. (The same name as in the previous command.
- `<target>` is a file path starting with local: as the prefix, followed by `<archiveName>`.

Execution of the export command saves the exported archive file on the file system of your NetScaler appliance in the /var/tmp folder.

Examples:

```
> archive appfw profile test_pr archived_test_pr
```

```
> export appfw archive archived_test_pr local:dutA_test_pr
```

After the above two commands are executed, the /var/tmp folder contains the archived_test_pr file and the exported copy, dutA_test_pr, which are identical in size. From the CLI, you can drop into the shell to navigate to the folder to verify that these files are there.

After exporting the archive file, you can use **scp** or some other such file transfer utility to transfer a copy of the archive file from the NetScaler appliance on which they were created to your target NetScaler appliance.

Importing App Firewall profiles with the CLI

After you have successfully scp'd the archived file from the source appliance to the target appliance, you are ready to **import** the profile's archive, and then execute the **restore** command to replicate the profile's configuration on the target appliance.

Log onto the target appliance. Drop into the shell and cd to the /var/tmp folder to verify that the size of the scp'd file on this appliance matches the size of the original archived file on the source appliance. Exit the shell to return to the command line.

To import a profile by using the CLI:

At the command prompt, type the following commands:

```
import appfw archive <src> <name> [-comment <string>]
```

where

- `<src>` is the location of the archive file after it has been transferred from the source appliance on which it was created. You can use a local file system and file name. If you have placed the archive on a server, you can use a URL to import the archived file. If the path or file name contains spaces, enclose the URL in straight double quotation marks.
- `<name>` is the name of the archive file to be imported.
- `<string>` is an optional description of the purpose of the Archive.

restore appfw profile `<archiveName>`

Examples:

A. Import from local file followed by restore:

```
> import appfw archive local:dutA_test_pr dut2_test_pr
```

```
> restore appfw profile dut2_test_pr
```

B. Import from URL followed by restore:

```
import appfw archive http://10.217.30.16/FFC/Profile_ImportExport/
dutA_test_pr.tgz my_archive
restore appfw profile my_archive
```

This example restores the test_pr profile along with all bound objects (such as signatures, html error page, relaxation rules and so on) on the target NetScaler appliance.

You can use the following CLI commands to access man pages for additional details.

- man archive appfw profile
- man export appfw archive
- man import appfw archive
- man restore appfw profile
- man show appfw archive
- man rm appfw archive

Exporting and importing App Firewall profiles with the GUI

The GUI is easier to use than the CLI. The utility performs both archive and export operations when you click **Export**. Similarly it executes both import and restore when you click **Import**. The GUI can access the local file system of the computer from which you access the utility. You can export a copy of the archive and save it on your local computer. You can then import this copy directly in the target appliance without having to manually transfer the archive file from one appliance to the other(s).

To export an App Firewall profile by using the GUI:

1. Navigate to **Configuration** > **Security** > **App Firewall** > **Profiles**.
2. In the details pane, select a profile to export. Click **Actions** and select **Export** to download and save a copy in your computer's local file system.

To import an App Firewall profile by using the GUI:

1. Navigate to **Configuration** > **Security** > **App Firewall** > **Profiles**.
2. In the details pane, click **Actions** and select **Import**. In the Import App Firewall Profile pane, the Import From* selection box gives you 2 options:

URL: You can choose to import an archive by specifying a **URL**. When this option is selected, you must provide an absolute path for the archived file in the **URL** input box.

File: You can choose to import an archive from the local **File**. When this option is selected, a **Local File** selection field is displayed. You can browse your computer's local files to select the target archive file.

Click **Create** to import the specified archive. Successful completion of the import operation creates the profile configuration on the target appliance.

Highlights

- You can replicate the entire configuration (including all import objects as well as configured relaxation rules for the profile) on multiple appliances, without needing to repeat configuration steps, by using export and import profile functionality.
- The imported objects, such as signatures, WSDL, Schema, error page and so on, are included in the archived tar file and replicated on the target appliance.
- Customized field types are included in the archived tar file and replicated on the target appliance.
- The policy bindings of the archived profile are not replicated when the configuration is restored. You must configure the policy and bind it to the profile after importing the profile to the appliance.
- The name of the archive file can be up to 31 character long. As with profile names, an archive name must begin with an alphanumeric character or underscore and contain only alphanumeric and underscore (_), number (#), period (.), space ( ), colon (:), at (@), equals (=) or hyphen (-) characters.
- Comments associated with the archive should be descriptive enough to convey the purpose of the archived configuration. The maximum allowed length for a comment is 255 characters.
- The "clear config –force basic" command does not remove the archived profiles.
- The import and export profile functionality is supported in high availability (HA) deployments.

Debugging tips

- Monitor the /var/log/ns.log during command executions to see if there are any ERROR messages.
- Additional logs (_restore.log, remove.log, import.log) are generated in the /var/tmp/folder. They can help debug issues during the corresponding operations. When these logs reach one MB in size, the log messages are purged to shrink the log file to one fourth of the original size.
- If the import command fails when you are using the URL option instead of the local file system, verify that DNS name server and route settings are accurately configured.
- If you are using the HTTPS protocol to import the archive, the command might fail if the HTTPS server requires client certificate authentication.

Configuring and using the Learning feature

September 21, 2020

The learning feature is a repetitive pattern filter that observes activity on a web site or application protected by the App Firewall, to determine what constitutes normal activity on that web site or application. It then generates a list of up to 2,000 suggested rules or exceptions (relaxations) for each security checks that includes support for the learning feature. Users normally find it easier to configure relaxations by using the learning feature than by entering the necessary relaxations manually.

The security checks that support the learning feature are:

- Start URL check
- Cookie Consistency check
- Form Field Consistency check
- Field Formats check
- CSRF Form Tagging check
- HTML SQL Injection check
- HTML Cross-Site Scripting check
- XML Denial-of-Service check
- XML Attachment check
- Web Services Interoperability check

You perform two different types of activities when using the learning feature. First, you enable and configure the feature to use it. You can use learning on all traffic to your protected web applications, or you can configure a list of IPs (called the *Add Trusted Learning Clients* list) from which the learning feature should generate recommendations. Second, after the feature has been enabled and has processed a certain amount of traffic to your protected web sites, you review the list of suggested rules and relaxations (learned rules) and mark each with one of the following designations:

- **Edit & Deploy.** The rule is pulled into the Edit dialog box so that you can modify it, and the modified form is deployed.
- **Deploy.** The unmodified learned rule is placed on the list of rules or relaxations for this security check.
- **Skip.** The learned rule is placed on a list of rules or relaxations that are not deployed. The learned rule is removed when skipped. However, as they are not added to relaxations, they might get learned again.

Learning is not performed only when relaxations are in place, except for field format rules. When rules are skipped, they are only removed from learned database. As relaxations are not added, they might get learned again. When rules are deployed, they are removed from learned database and also relaxations are added for the rules. As relaxations are added, they would not be learned again. For fieldformat protection, learning is performed irrespective of relaxations.

Although you can use the command line interface for basic configuration of the learning feature, the feature is designed primarily for configuration through the App Firewall wizard or the GUI. You can perform only limited configuration of the learning feature by using the command line.

The wizard integrates configuration of learning features with configuration of the App Firewall as a whole, and is therefore the easiest method for configuring this feature on a new NetScaler appliance or when managing a simple App Firewall configuration. The GUI visualizer and manual interface both provide direct access to all learned rules for all security checks, and are therefore often preferable when you must review learned rules for a large number of security checks.

The learning database is limited to 20 MB in size, which is reached after approximately 2,000 learned rules or relaxations are generated per security check for which learning is enabled. If you do not regularly review and either approve or ignore learned rules and this limit is reached, an error is logged to the NetScaler log and no more learned rules are generated until you review the existing learned rules and relaxations.

If learning stops because the database has reached its size limit, you can restart learning either by reviewing the existing learned rules and relaxations or by resetting the learning data. After learned rules or relaxations are approved or ignored, they are removed from the database. After you reset the learning data, all existing learning data is removed from the database and it is reset to its minimum size. When the database falls below 20 MB in size, learning restarts automatically.

To configure the learning settings by using the command line interface

Specify the App Firewall profile to be configured and, for each security check that you want to include in that profile, specify the minimum threshold or the percent threshold. The minimum threshold is an integer representing the minimum number of user sessions that the App Firewall must process before it learns a rule or relaxation (default: 1). The percent threshold is an integer representing the percentage of user sessions in which the App Firewall must observe a particular pattern (URL, cookie, field, attachment, or rule violation) before it learns a rule or relaxation (default: 0). Use the following commands:

- `set appfw learningsettings <profileName> [-startURLMinThreshold <positive_integer>] [-startURLPercentThreshold <positive_integer>] [-cookieConsistencyMinThreshold <positive_integer>] [-cookieConsistencyPercentThres <positive_integer>] [-CSRFtagMinThreshold <positive_integer>] [-CSRFtagPercentThreshold <positive_integer>] [-fieldConsistencyMinThreshold <positive_integer>] [-fieldConsistencyPercentThreshold <positive_integer>] [-crossSiteScriptingMinThreshold <positive_integer>] [-crossSiteScriptingPerce <positive_integer>] [-SQLInjectionMinThreshold <positive_integer>] [-SQLInjectionPercentThreshold <positive_integer>] [-fieldFormatMinThreshold <positive_integer>] [-fieldFormatPercentThreshold <positive_integer>] [-XMLWSIMinThreshold <positive_integer>] [-XMLWSIPercentThreshold <positive_integer>] [-XMLAttachmentMinThreshold <positive_integer>] [-XMLAttachmentPercentThreshold <positive_integer>]`
- `save ns config`

Example

The following example enables and configures the learning settings in the profile pr-basic for the HTML SQL Injection security check. This is an appropriate initial test bed learning configuration, where you have complete control over the traffic that is sent to the App Firewall.

```
set appfw learningsettings pr-basic -SQLInjectionMinThreshold 10
set appfw learningsettings pr-basic -SQLInjectionPercentThreshold 70
save ns config
<!--NeedCopy-->
```

To reset learning settings to their defaults by using the command line interface

To remove any custom configuration of the learning settings for the specified profile and security check, and return the learning settings to their defaults, at the command prompt type the following commands:

- `unset appfw learningsettings <profileName> [-startURLMinThreshold] [-startURLPercentThreshold] [-cookieConsistencyMinThreshold] [-cookieConsistencyPercentThreshold] [-CSRFtagMinThreshold] [-CSRFtagPercentThresh] [-fieldConsistencyMinThreshold] [-fieldConsistencyPercentThreshold] [-crossSiteScriptingMinThreshold] [-crossSiteScriptingPercentThreshold] [-SQLInjectionMinThreshold] [-SQLInjectionPercentThreshold] [-`

```
fieldFormatMinThreshold] [-fieldFormatPercentThreshold ] [-XMLWSIMinThreshold
 ] [-XMLWSIPercentThreshold ] [-XMLAttachmentMinThreshold ] [-XMLAttachmentPercen
]
```

- `save ns config`

To display the learning settings for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningsettings <profileName>
```

To display unreviewed learned rules or relaxations for a profile by using the command line interface

At the command prompt, type the following command:

```
show appfw learningdata <profileName> <securityCheck>
```

To remove specific unreviewed learned rules or relaxations from the learning database by using the command line interface

At the command prompt, type the following command:

```
rm appfw learningdata <profileName> (-startURL <expression> | -cookieConsistency
 <string> | (-fieldConsistency <string> <formActionURL>)| (-crossSiteScripting
 <string> <formActionURL>)| (-SQLInjection <string> <formActionURL>)| (-
fieldFormat <string><formActionURL>)| (-CSRFTag <expression> <CSRFFormOriginURL
>)| -XMLDoSCheck <expression> | -XMLWSICheck <expression> | -XMLAttachmentCheck
 <expression>)[-TotalXMLRequests]
```

Example

The following example removes all unreviewed learned relaxations for the pr-basic profile, HTML SQL Injection security check, that apply to the LastName form field.

```
rm appfw learningdata pr-basic -SQLInjection LastName
<!--NeedCopy-->
```

To remove all unreviewed learned data by using the command line interface

At the command prompt, type the following command:

```
reset appfw learningdata
```

To export learning data by using the command line interface

At the command prompt, type the following command:

```
export appfw learningdata <profileName> <securitycheck>[-target <string>]
```

Example

The following example exports learned relaxations for the pr-basic profile and the HTML SQL Injection security check to a comma-separated values (CSV) format file in the /var/learnt_data/ directory under the filename specified in the -target parameter.

```
export appfw learningdata pr-basic SQLInjection -target sqli_ld
<!--NeedCopy-->
```

To configure the Learning feature by using the GUI

1. Navigate to **Security** > **App Firewall** > **Profiles**.
2. In the **Profiles** pane, select the profile, and then click **Edit**.
3. Click the Learning tab. At the top of the Learning tab is list of the security checks that are available in the current profile and that support the learning feature.
4. To configure the learning thresholds, select a security check, and then type the appropriate values in the following text boxes:
   Minimum number threshold. Depending on which security check's learning settings you are configuring, the minimum number threshold might refer to the minimum number of total user sessions that must be observed, the minimum number of requests that must be observed, or the minimum number of times a specific form field must be observed, before a learned relaxation is generated. Default: 1

   - **Percentage of times threshold.** Depending on which security check's learning settings you are configuring, the percentage of times threshold might refer to the percentage of total observed user sessions that violated the security check, the percentage of requests, or the percentage of times a form field matched a particular field type, before a learned relaxation is generated. Default: 0

5. To remove all learned data and reset the learning feature, so that it must start its observations again from the beginning, click **Remove All Learned Data**.
   Note: This button removes only learned recommendations that have not been reviewed and either approved or skipped. It does not remove learned relaxations that have been accepted and deployed.

6. To restrict the learning engine to traffic from a specific set of IPs, click **Trusted Learning Clients**, and add the IP addresses that you want to use to the list.
    a) To add an IP address or IP address range to the Trusted Learning Clients list, click **Add**.
    b) In the **Add Trusted Learning Clients** dialog box, Trusted Clients IP list box, type the IP address or an IP address range in CIDR format.
    c) In the Comments text area, type a comment that describes this IP address or range.
    d) Click **Create** to add your new IP address or range to the list.
    e) To modify an existing IP address or range, click the IP address or range, and then click **Open**. Except for the name, the dialog box that appears is identical to the **Add Trusted Learning Clients** dialog box.
    f) To disable or enable an IP address or range, but leave it on the list, click the IP address or range, and then click **Disable or Enable**, as appropriate.
    g) To remove an IP address or range completely, click the IP address or range, and then click **Remove**.
7. Click **Close** to return to the Configure App Firewall Profile dialog box.
8. Click **Close** to close the **Configure App Firewall Profile** dialog box, and return to the **App Firewall Profile** screen.

To review learned rules or relaxations by using the GUI

1. Navigate to **Security** > **App Firewall** > **Profiles**.
2. Select the security check for which you want to review learned rules or relaxations, and then click **Manage Rules**.
3. In the **Manage Learned Rules** dialog box, choose how you want to review the learned rules.
    - To review the actual learned patterns as displayed in the window, do nothing and proceed to the next step.
    - To review the learned data hierarchically as a branching tree, enabling you to choose general patterns that match many of the learned patterns, click** Visualizer**.
4. If you have chosen to review actual learned patterns, perform the following steps.
5. Select the first learned relaxation and choose how to handle it.
   - To modify and then accept the relaxation, click **Edit & Deploy**, edit the relaxation regular expression, and then click **OK**.
   - To accept the relaxation without modifications, click **Deploy**.
   - To remove the relaxation from the list without deploying it, click **Skip**.
    a) Repeat the previous step to review each additional learned relaxation.
6. If you have chosen to use the Learning Visualizer, perform the following steps.

   a) In the branching hierarchical display, select a node that contains a learned pattern, and choose how to handle it.

      The screen area beneath the tree structure, under Regex of Selected Node, displays a generalized expression that matches all of the patterns in that node. If you want to display an expression that matches just one of the branches or just one of the leaves, select that branch or leaf.

      - To modify and then accept the learned relaxation, click Edit & Deploy, edit the relaxation regular expression, and then click **OK**.
      - To accept the relaxation without modifications, click Deploy.
      - To remove the modification from the list without deploying it, click **Skip**.

   b) Repeat the previous step to review other portions of the display.

7. Click **Close** to return to the **Manage Learned Rules** dialog box.
8. Click **Close** to return to the **Configure App Firewall Profile** dialog box.
9. Click **Close** to close the **Configure App Firewall Profile** dialog box, and return to the **App Firewall Profile** screen.

Supplemental Information about profiles

September 21, 2020

Following is supplemental information about particular aspects of App Firewall profiles. This information explains how to include special characters in a security check rule or relaxation, and how to use variables when configuring profiles.

Configuration variable support

Instead of using static values, to configure the App Firewall's security checks and settings, you can now use standard NetScaler named variables. By creating variables, you can more easily export and then import configurations to new NetScaler appliances, or update existing NetScaler appliances from a single set of configuration files. This simplifies updates when you use a test bed setup to develop a complex App Firewall configuration that is tuned for your local network and servers and then transfer that configuration to your production NetScaler appliances.

You create App Firewall configuration variables in the same manner as you do any other NetScaler named variables, following standard NetScaler conventions. To create a named expression variable by using the GUI, you use the ‘Add Expression dialog box. To create a named expression variable by using the NetScaler command line, you use the add expression command followed by the appropriate parameter.

The following URLs and expressions can be configured with variables instead of static values:

- **Start URL** (-starturl)
- **Deny URL** (-denyurl)
- **Form Action URL** for *Form Field Consistency Check* (-fieldconsistency)
- **Action URL** for *XML SQL Injection Check* (-xmlSQLInjection)
- **Action URL** for *XML Cross-Site Scripting Check* (-xmlXSS)
- **Form Action URL** for *HTML SQL Injection Check* (-sqlInjection)
- **Form Action URL** for *Field Format Check* (-fieldFormat)
- **Form Origin URL** and **Form Action URL** for *Cross-Site Request Forgery (CSRF) Check* (-csrfTag)
- **Form Action URL** for *HTML Cross-Site Scripting Check* (-crossSiteScripting)
- **Safe Object** (-safeObject)
- **Action URL** for *XML Denial-of-Service (XDoS) check* (-XMLDoS)
- **URL** for *Web Services Interoperability check* (-XMLWSIURL)
- <**URL** for *XML Validation check* (-XMLValidationURL)
- **URL** for *XML Attachment check* (-XMLAttachmentURL)

For more information, see "Policies and Expressions."

To use a variable in the configuration, you enclose the variable name between two at (@) symbols and then use it exactly as you would the static value that it replaces. For example, if you are configuring the Deny URL check by using the GUI and want to add the named expression variable myDenyURL to the configuration, you would type @myDenyURL@ into the Add Deny URL dialog box, Deny URL text area. To do the same task by using the NetScaler command line, you would type add appfw profile <`name`> -denyURLAction @myDenyURL@.

PCRE character encoding format

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your App Firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your App Firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.

- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.
- **ASCII control characters.** Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, "English US," the App Firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The App Firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The App Firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The App Firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The App Firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The App Firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The App Firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The App Firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the App Firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the App Firewall configuration is “xNN” for ASCII characters; “\xNN\xNN” for non-ASCII characters used in English, Russian, and Turkish; and “\xNN\xNN\xNN” for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an App Firewall regular expression as a UTF-8 character, you would type \x21. If you want to include an á, you would type \xC3\xA1.

> **Note**:
> Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character’s UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as x20 to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the App Firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters.

  Actual URL: `http://www.josénuñez.com`

  Encoded URL: `^http://www\[.\]jos\xC3\xA9nu\xC3\xB1ez\[.\]com$`

- Another URL containing extended ASCII characters.

  Actual URL: `http://www.example.de/trömso.html`

  Encoded URL: `^http://www[.]example\[.]de/tr\xC3\xB6mso[.]html$`

- A form field name containing extended ASCII characters.

  `Actual Name: nome_do_usuário`

  Encoded Name: `^nome_do_usu\xC3\xA1rio$`

- A safe object expression containing extended ASCII characters.

  Unencoded Expression [A-Z]{3,6}¥[1-9][0-9]{6,6}

  Encoded Expression: [A-Z]{3,6}\xC2\xA5[1-9][0-9]{6,6}

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

```
http://www.utf8-chartable.de/unicode-utf8-table.pl
<!--NeedCopy-->
```

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Inverted PCRE Expressions

In addition to matching content that contains a pattern, you can match content that does not contain a pattern by using an inverted PCRE expression. To invert an expression, you simply include an exclamation point (!) followed by white space as the first character in the expression.

> **Note**: If an expression consists only of an exclamation point with nothing following, the exclamation point is treated as a literal character, not syntax indicating an inverted expression.

The following App Firewall commands support inverted PCRE expressions:

- Start URL (URL)
- Deny URL (URL)
- Form Field Consistency (form action URL)
- Cookie Consistency (form action URL)
- Cross Site Request Forgery (CSRF) (form action URL)
- HTML Cross-site Scripting (form action URL)
- Field Format (form action URL)
- Field Type (type)
- Confidential Field (URL)

Note: If the security check contains an isRegex flag or check box, it must be set to YES or checked to enable regular expressions in the field. Otherwise the contents of that field are treated as literal and no regular expressions (inverted or not) are parsed.

Disallowed names for App Firewall profiles

The following names are assigned to built-in actions and profiles on the NetScaler appliance, and cannot be used as names for a user-created App Firewall profile.

- AGRESSIVE
- ALLOW
- BASIC
- CLIENTAUTH
- COMPRESS
- CSSMINIFY

- DEFLATE
- DENY
- DNS-NOP
- DROP
- GZIP
- HTMLMINIFY
- IMGOPTIMIZE
- JSMINIFY
- MODERATE
- NOCLIENTAUTH
- NOCOMPRESS
  - NONE
- NOOP
- NOREWRITE
- RESET
- SETASLEARNNSLOG_ACT
- SETNSLOGPARAMS_ACT
- SETSYSLOGPARAMS_ACT
- SETTMSESSPARAMS_ACT
  - SETVPNPARAMS_ACT
- SET_PREAUTHPARAMS_ACT
- default_DNS64_action
- dns_default_act_Cachebypass
- dns_default_act_Drop
- nshttp_default_profile
- nshttp_default_strict_validation
- nstcp_default_Mobile_profile
- nstcp_default_XA_XD_profile
- nstcp_default_profile
- nstcp_default_tcp_interactive_stream

- nstcp_default_tcp_lan
- nstcp_default_tcp_lan_thin_stream
- nstcp_default_tcp_lfp
- nstcp_default_tcp_lfp_thin_stream
- nstcp_default_tcp_lnp
- nstcp_default_tcp_lnp_thin_stream
- nstcp_internal_apps

Policy labels

September 22, 2020

A policy label consists of a set of policies, other policy labels, and virtual server-specific policy banks. The App Firewall evaluates each policy bound to the policy label in order of priority. If the policy matches, it filters the connection as specified in the associated profile. Then it does whatever the Goto parameter specifies, which can be to terminate policy evaluation, go to the next policy, or go to the policy with the specified priority. If the Invoke parameter is set, it terminates processing of the current policy label and begins to process the specified policy label or virtual server.

To create an App Firewall policy label by using the command line

At the command prompt, type the following commands:

- `add appfw policylabel <labelName> http_req`
- `save ns config`

Example

The following example creates a policy label named policylbl1.

```
add appfw policylabel policylbl1 http_req
save ns config
<!--NeedCopy-->
```

To bind a policy to a policy label by using the command line

At the command prompt, type the following commands:

- `bind appfw policylabel <labelName> <policyName> <priority> [<gotoPriorityExpressi >] [-invoke (<labelType> <labelName>)]`
- `save ns config`

Example

The following example binds the policy policy1 to the policy label policylbl1 with a priority of 1.

```
bind appfw policylabel policylbl1 policy1 1
save ns config
<!--NeedCopy-->
```

To configure an App Firewall policy label by using the GUI

1. Navigate to **Security > Citrix Web App Firewall > Policy Labels**.
2. In the details pane, do one of the following:

   - To add a new policy label, click **Add**.
   - To configure an existing policy label, select the policy label and the click **Open**.

   The **Create App Firewall Policy Label** or the **Configure App Firewall Policy Label** dialog box opens. The dialog boxes are nearly identical.

3. If you are creating a new policy label, in the Create App Firewall Policy Label dialog box, type a name for your new policy label.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

4. Select **Insert Policy** to insert a new row and display a drop-down list with all existing App Firewall policies.
5. Select the policy you want to bind to the policy label, or select New Policy to create a new policy and follow the instructions in To create and configure a policy by using the GUI. The policy that you selected or created is inserted into the list of globally bound App Firewall policies.
6. Make any additional adjustments.

   - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
   - To modify the policy expression, double click that field to open the Configure App Firewall Policy dialog box, where you can edit the policy expression.
   - To set the Goto Expression, double click field in the Goto Expression column heading to display the drop-down list, where you can choose an expression.

   - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression

7. Repeat steps 5 through 7 to bind any additional App Firewall policies you want to the policy label.
8. Click **Create** or **OK**, and then click **Close**. A message appears in the status bar, stating that you have successfully created or modified the policy label.

Policies

September 21, 2020

The App Firewall uses two types of policies: firewall policies and auditing policies. Firewall policies control which traffic is sent to the App Firewall. Auditing policies control the log server to which App Firewall logs are sent.

Firewall policies can be complex because the policy rule can consist of multiple expressions in the NetScaler expressions language, which is a full-fledged object oriented programming language capable of defining with extreme precision exactly which connections to filter. Because firewall policies operate within the context of the App Firewall, they must meet certain criteria that are connected to how the App Firewall functions and what traffic is appropriately filtered by it. As long as you keep these criteria in mind, however, firewall policies are similar to policies for other NetScaler features. The instructions here do not attempt to cover all aspects of writing firewall policies, but only provide an introduction to policies and cover those criteria that are unique to the App Firewall.

Auditing policies are simple because the policy rule is always ns_true. You need only specify the log server that you want to send logs to, the logging levels that you want to use, and a few other criteria that are explained in detail.

App Firewall Policies

September 21, 2020

A firewall policy is a rule associated with a profile. The rule is an expression or group of expressions that define the types of request/response pairs that the App Firewall is to filter by applying the profile. Firewall policy expressions are written in the NetScaler expressions language, an object-oriented programming language with special features to support specific NetScaler functions. The profile is the set of actions that the App Firewall is to use to filter request/response pairs that match the rule.

Firewall policies enable you to assign different filtering rules to different types of web content. Not all web content is alike. A simple web site that uses no complex scripting and accesses and handles no private data might require only the level of protection provided by a profile created with basic defaults. Web content that contains JavaScript-enhanced web forms or accesses an SQL database probably requires more tailored protection. You can create a different profile to filter that content and create a separate firewall policy that can determine which requests are attempting to access that content. You then associate the policy expression with a profile you created and globally bind the policy to put it into effect.

The App Firewall processes only HTTP connections, and therefore uses a subset of the overall NetScaler expressions language. The information here is limited to topics and examples that are likely to be useful when configuring the App Firewall. Following are links to additional information and procedures for firewall policies:

- For procedures that explain how to create and configure a policy, see Creating and Configuring App Firewall Policies.
- For a procedure that explains in detail how to create a policy rule (expression), see To create or configure an App Firewall rule (expression).
- For a procedure that explains how to use the Add Expression dialog box to create a policy rule, see To add a firewall rule (expression) by using the Add Expression dialog box.
- For a procedure that explains how to view the current bindings for a policy, see Viewing a Firewall Policy's Bindings.
- For procedures that explain how to bind an App Firewall policy, see Binding App Firewall Policies.
- For detailed information about the NetScaler expressions language, see Policies and Expressions.

> **Note**
>
> App Firewall evaluates the policies based on the configured priority and goto expressions. At the end of the policy evaluation, the last policy that evaluates to true is used and the security configuration of the corresponding profile is invoked for processing the request.
>
> For example, Consider a scenario where there are 2 policies.
>
> - Policy_1 is a generic policy with Expression=ns_true and has a corresponding profile_1 which is a basic profile. The priority is set to 100.
> - Policy_2 is more specific with Expression=HTTP.REQ.URL.CONTAINS("XYZ") and has a corresponding profile_2 which is an advance profile. The GoTo Expression is set to NEXT and the priority is set to 95 which is a higher priority compared to Policy_1.
>
> In this scenario, if the target string "XYZ" is detected in the URL of the processed request, Policy_2 match is triggered as it has a higher priority even though Policy_1 is also a match. However, as

> per the GoTo expression configuration of Policy_2, the policy evaluation continues and the next policy Policy_1 is also processed. At the end of the policy evaluation, Policy_1 evaluates as true and the basic security checks configured in Profile_1 are invoked.
>
> If the Policy_2 is modified and the GoTo Expression is changed from
> **NEXT** to
> **END**, the processed request that has the target string "XYZ", triggers the Policy_2 match due to priority consideration and as per the GoTo expression configuration, the policy evaluation ends at this point. Policy_2 evaluates as true and the advanced security checks configured in Profile_2 are invoked.
>
> **NEXT**
> **END**
>
> Policy evaluation is completed in one pass. Once the policy evaluation is completed for the request and the corresponding profile actions are invoked, the request does not go through another round of policy evaluation.

Creating and configuring App Firewall policies

September 21, 2020

A firewall policy consists of two elements: a *rule*, and an associated *profile*. The rule selects the HTTP traffic that matches the criteria that you set, and sends that traffic to the App Firewall for filtering. The profile contains the filtering criteria that the App Firewall uses.

The policy rule consists of one or more expressions in the NetScaler expressions language. The NetScaler expressions syntax is a powerful, object-oriented programming language that enables you to precisely designate the traffic that you want to process with a specific profile. For users who are not completely familiar with the NetScaler expressions language syntax, or who prefer to configure their NetScaler appliance by using a web-based interface, the GUI provides two tools: the Prefix menu and the Add Expression dialog box. Both help you to write expressions that select exactly the traffic that you want to process. Experienced users who are thoroughly familiar with the syntax may prefer to use the NetScaler command line to configure their NetScaler appliances.

> **Note**: In addition to the default expressions syntax, for backward compatibility the NetScaler operating system supports the NetScaler classic expressions syntax on NetScaler Classic and nCore appliances and virtual appliances. Classic expressions are not supported on NetScaler Cluster appliances and virtual appliances. Current NetScaler users who want to migrate existing configurations to the NetScaler Cluster must migrate any policies that contain classic expressions to the default expressions syntax.

For detailed information about the NetScaler expressions languages, see "Policies and Expressions."

You can create a firewall policy by using the GUI or the NetScaler command line.

To create and configure a policy by using the command line interface

At the command prompt, type the following commands:

- `add appfw policy <name><rule> <profileName>`
- `save ns config`

Example

The following example adds a policy named pl-blog, with a rule that intercepts all traffic to or from the host blog.example.com, and associates that policy with the profile pr-blog. This is an appropriate policy to protect a blog hosted on a specific hostname.

```
add appfw policy pl-blog "HTTP.REQ.HOSTNAME.DOMAIN.EQ("blog.example.com
   ")" pr-blog
<!--NeedCopy-->
```

To create and configure a policy by using the GUI

1. Navigate to **Security** > **App Firewall** > **Policies**.
2. In the details pane, do one of the following:

    - To create a new firewall policy, click **Add**. The **Create App Firewall Policy** is displayed.
    - To edit an existing firewall policy, select the policy, and then click **Edit**.

    The **Create App Firewall Policy** or **Configure App Firewall Policy** is displayed.

3. If you are creating a new firewall policy, in the **Create App Firewall Policy** dialog box, Policy Name text box, type a name for your new policy.

    The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 128 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), colon (:), and underscore (_) symbols.

    If you are configuring an existing firewall policy, this field is read-only. You cannot modify it.

4. Select the profile that you want to associate with this policy from the Profile drop-down list. You can create a new profile to associate with your policy by clicking New, and you can modify an existing profile by clicking Modify.
5. In the Expression text area, create a rule for your policy.

    - You can type a rule directly into the text area.

  - You can click Prefix to select the first term for your rule, and follow the prompts.
  - You can click Add to open the Add Expression dialog box, and use it to construct the rule.

6. Click **Create** or **OK**, and then click **Close**.

To create or configure an App Firewall rule

The policy rule, also called the *expression*, defines the web traffic that the App Firewall filters by using the profile associated with the policy. Like other NetScaler policy rules (or *expressions*), App Firewall rules use NetScaler expressions syntax. This syntax is powerful, flexible, and extensible. It is too complex to describe completely in this set of instructions. You can use the following procedure to create a simple firewall policy rule, or you can read it as an overview of the policy creation process.

1. If you have not already done so, navigate to the appropriate location in the **App Firewall** wizard or the NetScaler GUI to create your policy rule:
   - If you are configuring a policy in the **App Firewall** wizard, in the navigation pane, click **App Firewall**, then in the details pane click **App Firewall Wizard**, and then navigate to the **Specify Rule** screen.
   - If you are configuring a policy manually, in the navigation pane, expand **App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, the **Create App Firewall Profile** dialog box, or the **Configure App Firewall Profile** dialog box, click **Prefix**, and then choose the prefix for your expression from the drop-down list. Your choices are:
   - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol.
   - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
   - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.
   - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

   After you choose a prefix, the App Firewall displays a two-part prompt window that displays the possible next choices at the top, and a brief explanation of what the selected choice means at the bottom.
3. Choose your next term.

   If you chose HTTP as your prefix, your only choice is REQ, which specifies the Request/Response pair. (The App Firewall operates on the request and response as a unit instead of on each separately.) If you chose another prefix, your choices are more varied. For help on a specific choice, click that choice once to display information about it in the lower prompt window.

When you have decided which term you want, double-click it to insert it into the **Expression** window.

4. Type a period after the term you just chose. You are then prompted to choose your next term, as described in the previous step. When a term requires that you type a value, fill in the appropriate value. For example, if you choose HTTP.REQ.HEADER(“”), type the header name between the quotation marks.
5. Continue choosing terms from the prompts and filling in any values that are needed, until your expression is finished.

   Following are some examples of expressions for specific purposes.

   - **Specific web host.** To match traffic from a particular web host:

     ```
     HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
     <!--NeedCopy-->
     ```

     For shopping.example.com, substitute the name of the web host that you want to match.

   - **Specific web folder or directory.** To match traffic from a particular folder or directory on a Web host:

     ```
     HTTP.REQ.URL.STARTSWITH("https//www.example.com/folder")
     <!--NeedCopy-->
     ```

     For www.example.com, substitute the name of the web host. For folder, substitute the folder or path to the content that you want to match. For example, if your shopping cart is in a folder called /solutions/orders, you substitute that string for folder.

   - **Specific type of content: GIF images.** To match GIF format images:

     ```
     HTTP.REQ.URL.ENDSWITH(".png")
     <!--NeedCopy-->
     ```

     To match other format images, substitute another string in place of .png.

   - **Specific type of content: scripts.** To match all CGI scripts located in the CGI-BIN directory:

     ```
     HTTP.REQ.URL.STARTSWITH("https//www.example.com/CGI-BIN")
     <!--NeedCopy-->
     ```

     To match all JavaScripts with .js extensions:

```
HTTP.REQ.URL.ENDSWITH(".js")
<!--NeedCopy-->
```

For more information about creating policy expressions, see "Policies and Expressions."

> **Note**: If you use the command line to configure a policy, remember to escape any double quotation marks within NetScaler expressions. For example, the following expression is correct if entered in the GUI:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

If entered at the command line, however, you must type this instead:

```
HTTP.REQ.HEADER("Host").EQ("shopping.example.com")
<!--NeedCopy-->
```

To add a firewall rule (expression) by using the Add Expression dialog box

The Add Expression dialog box (also referred to as the Expression Editor) helps users who are not familiar with the NetScaler expressions language to construct a policy that matches the traffic that they want to filter.

1. If you have not already done so, navigate to the appropriate location in the **App Firewall** wizard or the NetScaler GUI:
    - If you are configuring a policy in the **App Firewall** wizard, in the navigation pane, click **App Firewall**, then in the details pane click **App Firewall Wizard**, and then navigate to the **Specify Rule** screen.
    - If you are configuring a policy manually, in the navigation pane, expand **App Firewall**, then **Policies**, and then **Firewall**. In the details pane, to create a new policy, click **Add**. To modify an existing policy, select the policy, and then click **Open**.
2. On the **Specify Rule** screen, in the **Create App Firewall Profile** dialog box, or in the **Configure App Firewall Profile** dialog box, click **Add**.
3. In the **Add Expression dialog** box, in the Construct Expression area, in the first list box, choose one of the following prefixes:
    - **HTTP.** The HTTP protocol. Choose this if you want to examine some aspect of the request that pertains to the HTTP protocol. The default choice.
    - **SYS.** The protected Web site(s). Choose this if you want to examine some aspect of the request that pertains to the recipient of the request.
    - **CLIENT.** The computer that sent the request. Choose this if you want to examine some aspect of the sender of the request.

  - **SERVER.** The computer to which the request was sent. Choose this if you want to examine some aspect of the recipient of the request.

4. In the second list box, choose your next term. The available terms differ depending on the choice you made in the previous step, because the dialog box automatically adjusts the list to contain only those terms that are valid for the context. For example, if you selected HTTP in the previous list box, the only choice is REQ, for requests. Because the App Firewall treats requests and associated responses as a single unit and filters both, you do not need to specific responses separately. After you choose your second term, a third list box appears to the right of the second. The Help window displays a description of the second term, and the **Preview Expression** window displays your expression.
5. In the third list box, choose the next term. A new list box appears to the right, and the Help window changes to display a description of the new term. The **Preview Expression** window updates to display the expression as you have specified it to that point.
6. Continue choosing terms, and when prompted filling in arguments, until your expression is complete. If you make a mistake or want to change your expression after you have already selected a term, you can simply choose another term. The expression is modified, and any arguments or additional terms that you added after the term that you modified are cleared.
7. When you have finished constructing your expression, click **OK** to close the **Add Expression** dialog box. Your expression is inserted into the **Expression** text area.

Binding App Firewall policies

September 21, 2020

After you have configured your App Firewall policies, you bind them to Global or a bind point to put them into effect. After binding, any request or response that matches an App Firewall policy is transformed by the profile associated with that policy.

When you bind a policy, you assign a priority to it. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer. In the NetScaler OS, policy priorities work in reverse order - the higher the number, the lower the priority.

Because the App Firewall feature implements only the first policy that a request matches, not any additional policies that it might also match, policy priority is important for achieving the results that you intend. If you give your first policy a low priority (such as 1000), you configure the App Firewall to perform it only if other policies with a higher priority do not match a request. If you give your first policy a high priority (such as 1), you configure the App Firewall to perform it first, and skip any other policies that might also match. You can leave yourself plenty of room to add other policies in any order, without having to reassign priorities, by setting priorities with intervals of 50 or 100 between each policy when you bind your policies.

For more information about binding policies on the NetScaler appliance, see "Policies and Expressions."

To bind an App Firewall policy by using the command line interface

At the command prompt, type the following commands:

- `bind appfw global <policyName>`
- `save ns config`
- `bind appfw profile <profile_name> -crossSiteScripting data`

Example

The following example binds the policy named pl-blog and assigns it a priority of 10.

```
bind appfw global pl-blog 10 save ns config
<!--NeedCopy-->
```

To bind an App Firewall policy by using the GUI

1. Do one of the following:
    - Navigate to **Security** > **App Firewall**, and in the details pane, click App Firewall policy manager.
    - Navigate to **Security** > **App Firewall** > **Policies** > **Firewall Policies**, and in the details pane, click **Policy Manager**.
2. In the **App Firewall Policy Manager** dialog, choose the bind point to which you want to bind the policy from the drop-down list. The choices are:
    - **Override Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance, and are applied before any other policies.
    - **LB Virtual Server.** Policies that are bound to a load balancing virtual server are applied only to traffic that is processed by that load balancing virtual server, and are applied before any Default Global policies. After selecting LB Virtual Server, you must also select the specific load balancing virtual server to which you want to bind this policy.
    - **CS Virtual Server.** Policies that are bound to a content switching virtual server are applied only to traffic that is processed by that content switching virtual server, and are applied before any Default Global policies. After selecting CS Virtual Server, you must also select the specific content switching virtual server to which you want to bind this policy.
    - **Default Global.** Policies that are bound to this bind point process all traffic from all interfaces on the NetScaler appliance.

- **Policy Label.** Policies that are bound to a policy label process traffic that the policy label routes to them. The policy label controls the order in which policies are applied to this traffic.
- **None.** Do not bind the policy to any bind point.

3. Click **Continue**. A list of existing App Firewall policies appears.
4. Select the policy you want to bind by clicking it.
5. Make any additional adjustments to the binding.
    - To modify the policy priority, click the field to enable it, and then type a new priority. You can also select Regenerate Priorities to renumber the priorities evenly.
    - To modify the policy expression, double click that field to open the **Configure App Firewall Policy** dialog box, where you can edit the policy expression.
    - To set the Goto Expression, double click **field** in the Goto Expression column heading to display the drop-down list, where you can choose an expression.
    - To set the Invoke option, double click field in the Invoke column heading to display the drop-down list, where you can choose an expression
6. Repeat steps 3 through 6 to add any additional App Firewall policies you want to globally bind.
7. Click **OK**. A message appears in the status bar, stating that the policy has been successfully bound.

Viewing a policy bindings

September 22, 2020

You can quickly check to determine what bindings are in place for any firewall policy by viewing the bindings in the GUI.

To view bindings for an App Firewall policy

1. Navigate to **Security > Citrix Web App Firewall > Policies > Firewall Policies**
2. In the details pane, select the policy that you want to check, and then click Show Bindings. The Binding Details for Policy: Policy message box is displayed, with a list of bindings for the selected policy.
3. Click **Close**.

Supplemental information about App Firewall policies

September 21, 2020

Following is supplemental information about particular aspects of App Firewall policies that system administrators who manage the App Firewall might need to know.

Correct but unexpected behavior

Web application security and modern web sites are complex. In a number of scenarios, a NetScaler policy might cause the App Firewall to behave differently in certain situations than a user who is familiar with policies would normally expect. Following are a number of cases where the App Firewall may behave in an unexpected fashion.

- **Request with a missing HTTP Host header and an absolute URL.** When a user sends a request, in the majority of cases the request URL is relative. That is, it takes as its starting point the Referer URL, the URL where the user's browser is located when it sends the request. If a request is sent without a Host header, and with a relative URL, the request is normally blocked both because it violates the HTTP specification and because a request that fails to specify the host could under some circumstances constitute an attack. If a request is sent with an absolute URL, however, even if the Host header is missing, the request bypasses the App Firewall and is forwarded to the web server. Although such a request violates the HTTP specification, it poses no possible threat because an absolute URL contains the host.

Auditing policies

September 22, 2020

Auditing policies determine the messages that are generated and logged during an App Firewall session. These messages are logged in SYSLOG format to the local NSLOG server or to an external logging server. Different types of messages are logged on the basis of the level of logging selected.

To create an auditing policy, you must first create either an NSLOG server or a SYSLOG server. After specifying the server, you create the policy and specify the type of log and the server to which logs are sent.

To create an auditing server by using the command line interface

You can create two different types of auditing server: an NSLOG server or a SYSLOG server. The command names are different, but the parameters for the commands are the same.

To create an auditing server, at the command prompt, type the following commands:

- `add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-logFacility`

```
<logFacility>] [-tcp ( NONE | ALL )] [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME | LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )]
 [-appflowExport ( ENABLED | DISABLED )]
```

- `save ns config`

Example

The following example creates a syslog server named syslog1 at IP 10.124.67.91, with loglevels of emergency, critical, and warning, log facility set to LOCAL1, that logs all TCP connections:

```
add audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning -logFacility
LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

To modify or remove an auditing server by using the command line interface

- To modify an auditing server, type the set audit <`type`> command, the name of the auditing server, and the parameters to be changed, with their new values.
- To remove an auditing server, type the rm audit <`type`> command and the name of the auditing server.

Example

The following example modifies the syslog server named syslog1 to add errors and alerts to the log level:

```
set audit syslogAction syslog1 10.124.67.91 -logLevel emergency critical warning alert error
-logFacility LOCAL1 -tcp ALL
save ns config
<!--NeedCopy-->
```

To create or configure an auditing server by using the GUI

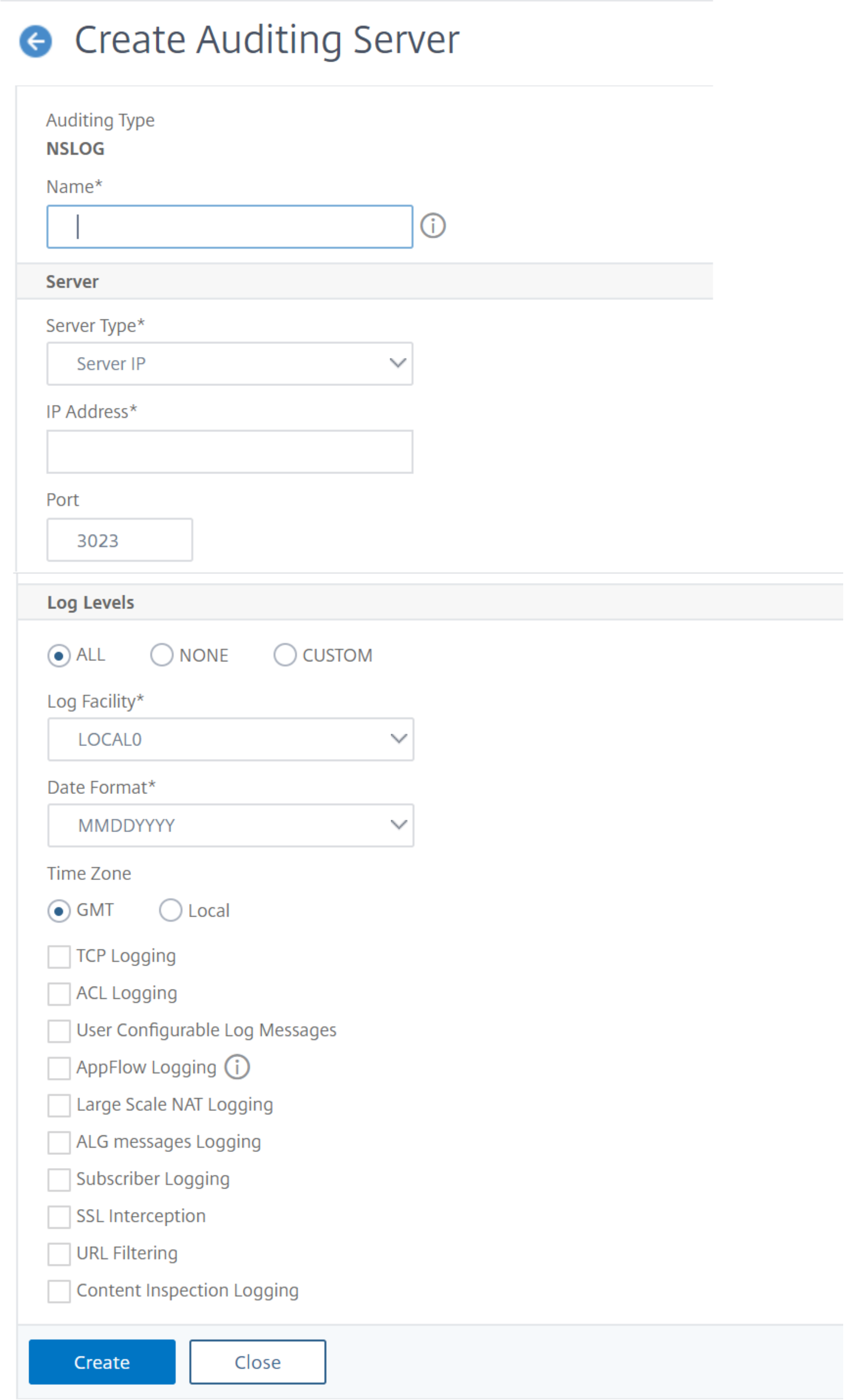
1. Navigate to **Security** > **Web App Firewall** > **Policies** > **Auditing** > **Nslog**.
2. In the Nslog Auditing page, click**Servers** tab.
3. Do one of the following:
    - To add a new auditing server, click **Add**.

   - To modify an existing auditing server, select the server, and then click **Edit**.

4. In the **Create Auditing Server** page, set the following parameters:

   - Name
   - Server Type
   - IP Address
   - Port
   - Log Levels
   - Log Facility
   - Date Format
   - Time Zone
   - TCP Logging
   - ACL Logging
   - User Configurable Log Messages
   - AppFlow Logging
   - Large Scale NAT Logging
   - ALG messages logging
   - Subscriber logging
   - SSL Interception
   - URL Filtering
   - Content Inspection Logging

5. Click **Create** and **Close**.



Create Auditing Server

Auditing Type
NSLOG

Name*

Server

Server Type*
Server IP

IP Address*

Port
3023

Log Levels

● ALL ○ NONE ○ CUSTOM

Log Facility*
LOCAL0

Date Format*
MMDDYYYY

Time Zone
● GMT ○ Local

☐ TCP Logging
☐ ACL Logging
☐ User Configurable Log Messages
☐ AppFlow Logging
☐ Large Scale NAT Logging
☐ ALG messages Logging
☐ Subscriber Logging
☐ SSL Interception
☐ URL Filtering
☐ Content Inspection Logging

To create an auditing policy by using the command line interface

You can create an NSLOG policy or a SYSLOG policy. The type of policy must match the type of server. The command names for the two types of policy are different, but the parameters for the commands are the same.

At the command prompt, type the following commands:

- `add audit syslogPolicy <name> <-rule > <action>`
- `save ns config`

Example

The following example creates a policy named syslogP1 that logs App Firewall traffic to a syslog server named syslog1.

```
add audit syslogPolicy syslogP1 rule "ns_true" action syslog1
save ns config
<!--NeedCopy-->
```

To configure an auditing policy by using the command line interface

At the command prompt, type the following commands:

- `set audit syslogPolicy <name> [-rule <expression>] [-action <string>]`
- `save ns config`

Example

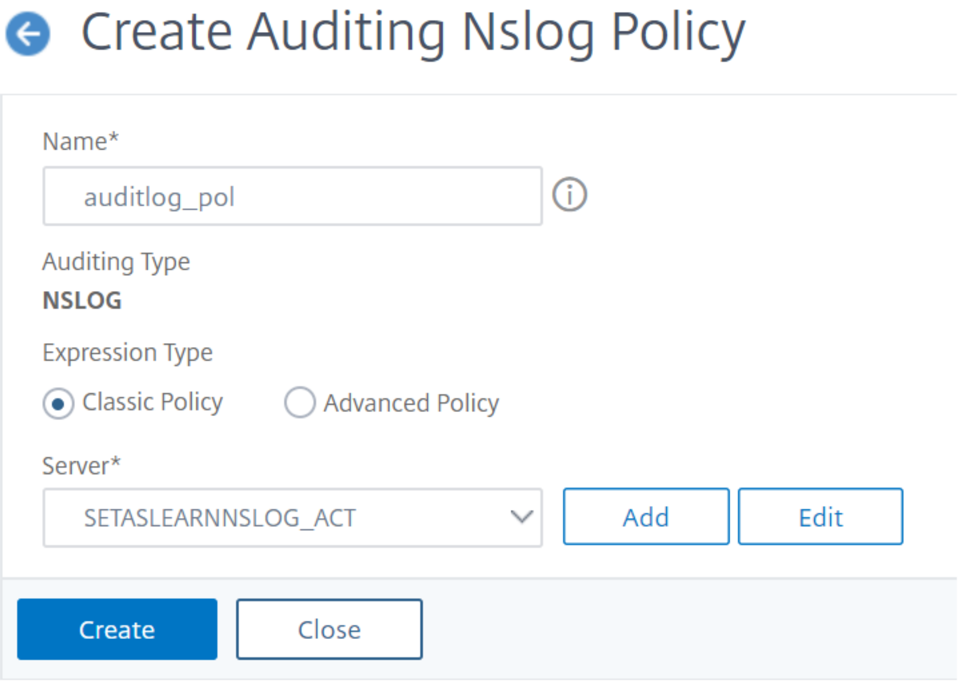
The following example modifies the policy named syslogP1 to log App Firewall traffic to a syslog server named syslog2.

```
set audit syslogPolicy syslogP1 rule "ns_true"action syslog2
save ns config
```

To configure an auditing policy by using the GUI

1. Navigate to **Security** > **Web App Firewall** > **Policies**.
2. In the details pane, click **Audit Nslog Policy**.
3. In the Nslog Auditing page, click **Policies** tab and do one of the following:
    - To add a new policy, click **Add**.
    - To modify an existing policy, select the policy, and then click **Edit**.

4. In the **Create Auditing Nslog Policy** page, set the following parameters:
   - Name
   - Auditing Type
   - Expresion Type
   - Server
5. Click **Create**.



Create Auditing Nslog Policy

Name*

auditlog_pol

Auditing Type

NSLOG

Expression Type

● Classic Policy ○ Advanced Policy

Server*

SETASLEARNNSLOG_ACT

Imports

September 21, 2020

Several App Firewall features make use of external files that you upload to the App Firewall when configuring it. Using the GUI, you manage those files in the Imports pane, which has four tabs corresponding to the four types of files you can import: HTML error objects, XML error objects, XML schemas, and Web Services Description Language (WSDL) files. Using the NetScaler command line, you can import these types of files, but you cannot export them.

HTML error object

When a user's connection to an HTML or Web 2.0 page is blocked, or a user asks for a non-existent HTML or Web 2.0 page, the App Firewall sends an HTML-based error response to the user's browser.

When configuring which error response the App Firewall should use, you have two choices:

- You can configure a redirect URL, which can be hosted on any Web server to which users also have access. For example, if you have a custom error page on your Web server, 404.html, you can configure the App Firewall to redirect users to that page when a connection is blocked.
- You can configure an HTML error object, which is an HTML-based Web page that is hosted on the App Firewall itself. If you choose this option, you must upload the HTML error object to the App Firewall. You do that in the Imports pane, on the HTML Error Object tab.

The error object must be a standard HTML file that contains no non-HTML syntax except for App Firewall error object customization variables. It cannot contain any CGI scripts, server-parsed code, or PHP code. The customization variables enable you to embed troubleshooting information in the error object that the user receives when a request is blocked. While most requests that the App Firewall blocks are illegitimate, even a properly configured App Firewall can occasionally block legitimate requests, especially when you first deploy it or after you make significant changes to your protected Web sites. By embedding information in the error page, you provide the user with the information that he or she needs to give to the technical support person so that any issues can be fixed.

The App Firewall error page customization variables are:

- ${NS_TRANSACTION_ID}. The transaction ID that the App Firewall assigned to this transaction.
- ${NS_APPFW_SESSION_ID}. The App Firewall session ID.
- ${NS_APPFW_VIOLATION_CATEGORY}. The specific App Firewall security check or rule that was violated.
- ${NS_APPFW_VIOLATION_LOG}. The detailed error message associated with the violation.
- ${COOKIE The contents of the specified cookie. For `<CookieName>`, substitute the name of the specific cookie that you want to display on the error page. If you have multiple cookies whose contents you want to display for troubleshooting, you can use multiple instances of this customization variable, each with the appropriate cookie name.
  Note: If you have blocking enabled for the Cookie Consistency Check, any blocked cookies are not displayed on the error page because the App Firewall blocks them.

To use these variables, you embed them in the HTML or XML of the error page object as if they were an ordinary text string. When the error object is displayed to the user, for each customization variable the App Firewall substitutes the information to which the variable refers. An example HTML error page that uses custom variables is shown below.

```
<!doctype html public "-//w3c//dtd html 4.0//en">  <html>  <head>  <
   title>Page Not Accessible</title>  </head>  <body>  <h1>Page Not
   Accessible</h1>  <p>The page that you accessed is not available. You
    can:</p>  <ul>  <li>return to the <b><a href="[homePage]">home page
   </a></b>, re-establish your session, and try again, or,</li>  <li>
   report this incident to the help desk via <b><a href="mailto:[
    helpDeskEmailAddress]">email</a></b> or by calling [
    helpDeskPhoneNumber].</li>  </ul>  <p>If you contact the help desk,
    please provide the following information:</p>  <table cellpadding=8
    width=80%>  <tr><th align="right" width=30%>Transaction ID:</th><td
    align="left" valign="top" width=70%>${
NS_TRANSACTION_ID }
</td></tr>  <tr><th align="right" width=30%>Session ID:</th><td align=
   "left" valign="top" width=70%>${
NS_APPFW_SESSION_ID }
</td></tr>  <tr><th align="right" width=30%>Violation Category:</th><
   td align="left" valign="top" width=70%>${
NS_APPFW_VIOLATION_CATEGORY }
</td></tr>  <tr><th align="right" width=30%>Violation Log:</th><td
   align="left" valign="top" width=70%>${
NS_APPFW_VIOLATION_LOG }
</td></tr>  <tr><th align="right" width=30%>Cookie Name:</th><td align
   ="left" valign="top" width=70%>${
COOKIE("[cookieName]") }
</td></tr>  </table>  <body>  <html>
<!--NeedCopy-->
```

To use this error page, copy it into a text or HTML editor. Substitute the appropriate local information for the following variables, which are enclosed in square brackets to distinguish them from the NetScaler variables. (Leave those unchanged.):

- `[homePage]`. The URL for your web site's home page.
- `[helpDeskEmailAddress]`. The email address that you want users to use to report blocking incidents.
- `[helpDeskPhoneNumber]`. The phone number that you want users to call to report blocking incidents.
- `[cookieName]`. The name of the cookie whose contents you want to display on the error page.

XML error object

When a user's connection to an XML page is blocked, or a user asks for a nonexistent XML application, the App Firewall sends an XML-based error response to the user's browser. You configure the error response by uploading an XML-based error page to the App Firewall in the Imports Pane, on the XML Error Object tab. All XML error responses are hosted on the App Firewall. You cannot configure a redirect URL for XML applications.

> **Note**:
> You can use the same customization variables in an XML error object as in an HTML error object.

XML Schema

When the App Firewall performs a validation check on a user's request for an XML or Web 2.0 application, it can validate the request against the XML schema or design type document (DTD) for that application and reject any request that does not follow the schema or DTD. Both an XML schema and a DTD are standard XML configuration files that describe the structure of a specific type of XML document.

WSDL

When the App Firewall performs a validation check on a user's request for an XML SOAP-based web service, it can validate the request against the web services type definition (WSDL) file for that web service. A WSDL file is a standard XML SOAP configuration file that defines the elements of a specific XML SOAP web service.

Importing and exporting files

September 22, 2020

You can import HTML or XML error objects, XML schemas, DTDs, and WSDLs to the App Firewall by using the GUI or the command line. You can edit any of these files in a web-based text area after importing them, to make small changes directly on the NetScaler instead of having to make them on your computer and then reimport them. Finally, you can export any of these files to your computer, or delete any of these files, by using the GUI.

> **Note**: You cannot delete or export an imported file by using the command line.

To import a file by using the command line interface

At the command prompt, type the following commands:

- `import appfw htmlerrorpage <src> <name>`
- `<save> ns config`

Example

The following example imports an HTML error object from a file named error.html and assigns it the name HTMLError.

```
import htmlerrorpage error.html HTMLError
save ns config
```

```
<!--NeedCopy-->
```

To import a file by using the GUI

Before you attempt to import an XML schema, DTD, or WSDL file, or an HTML or XML error object from a network location, verify that the NetScaler can connect to the Internet or LAN computer where the file is located. Otherwise, you cannot import the file or object.

1. Navigate to **Security > Citrix Web App Firewall > Imports**.
2. Navigate to **Application Firewall** > **Imports**.
3. In the **Application Firewall Imports** pane, select the tab for the type of file you want to import, and then click **Add**.

   The tabs are HTML Error Page, XML Error Page, XML Schema or WSDL. The upload process is identical on all four tabs from the user point of view.
4. Fill in the dialog fields.
   - **Name**—A name for the imported object.
   - **Import From**—Choose the location of the HTML file, XML file, XML schema or WSDL that you want to import in the drop-down list:
     - **URL:** A web URL on a website accessible to the appliance.
     - **File:** A file on a local or networked hard disk or other storage device.
     - **Text:** Type or paste the text of the custom response directly into a text field in the GUI.

     The third text box changes to the appropriate value. The three possible values are provided below.
   - **URL**—Type the URL into the text box.
   - **File**—Type the path and filename to the HTML file directly, or click Browse and browse to the HTML file.
   - **Text**—The third field is removed, leaving a blank space.
5. Click **Continue**. The File Contents dialog is displayed. If you chose URL or File, the File Contents text box contains the HTML file that you specified. If you chose Text, the File Contents text box is empty.
6. If you chose Text, type or copy and paste the custom response HTML that you want to import.
7. Click **Done**.
8. To delete an object, select the object, and then click **Delete**.

To export a file by using the GUI

Before you attempt to export an XML schema, DTD, or WSDL file, or an HTML or XML error object, verify that the App Firewall appliance can access the computer where the file is to be saved. Otherwise, you cannot export the file.

1. Navigate to **Security** > **App Firewall** > **Imports**.
2. In the **App Firewall Imports** pane, select the tab for the type of file you want to export.

   The export process is identical on all four tabs from the user point of view.
3. Select the file that you want to export.
4. Expand the Action drop-down list, and select **Export**.
5. In the dialog box, choose **Save File** and click **OK**.
6. In the **Browse** dialog box, navigate to the local file system and directory where you want to save the exported file, and click **Save**.

To edit an HTML or XML Error Object in the GUI

You edit the text of HTML and XML error objects in the GUI without exporting and then reimporting them.

1. Navigate to **Security > Citrix Web App Firewall > Imports**, and then select the tab for the type of file that you want to modify.
2. Navigate to **Application Firewall** > **Imports**, and then select the tab for the type of file that you want to modify.
3. Select the file that you want to modify, and then click **Edit**.

   The text of the HTML or XML error object is displayed in a browser text area. You can modify the text by using the standard browser-based editing tools and methods for your browser.

   Note: The edit window is designed to allow you to make minor changes to your HTML or XML error object. To make extensive changes, you may prefer to export the error object to your local computer and use standard HTML or XML web page editing tools.
4. Click **OK**, and then click **Close**.

Global configuration

September 21, 2020

The App Firewall global configuration affects all profiles and policies. The Global Configuration items are:

- **Engine Settings.** A collection of global settings—session cookie name, session time-out, maximum session lifetime, logging header name, undefined profile, default profile, and import size limit—that pertain to all connections that the App Firewall processes, rather than to a specific subset of connections.
- **Confidential Fields.** A set of form fields in web forms that contain sensitive information that should not be logged to the App Firewall logs. Form fields such as password fields on a logon page or credit card information on a shopping cart checkout form are normally designated as confidential fields.
- **Field Types.** The list of web form field types used by the Field Formats security check. Each of these field types is defined by a PCRE-compliant regular expression that defines the type of data and the minimum/maximum length of data that should be allowed in that type of form field.
- **XML Content Types.** The list of content types recognized as XML and subjected to XML-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.
- **JSON Content Types.** The list of content types recognized as JSON and subjected to JSON-specific security checks. Each of these content types is defined by a PCRE-compliant regular expression that defines the exact MIME type assigned to that content.

Engine settings

September 21, 2020

The engine settings affect all requests and responses that the App Firewall processes. They include the following items:

- **Cookie name**—The name of the cookie that stores the NetScaler session ID.
- **Session timeout**—The maximum inactive period allowed. If a user session shows no activity for this length of time, the session is terminated and the user is required to reestablish it by visiting a designated start page.
- **Cookie post-encrypts prefix**—The string that precedes the encrypted portion of any encrypted cookies.
- **Maximum session lifetime**—The maximum amount of time, in seconds, that a session is allowed to remain live. After this period is reached, the session is terminated and the user is required to reestablish it by visiting a designated start page. This setting cannot be less then the session timeout. To disable this setting, so that there is no maximum session lifetime, set the value to zero (0).
- **Logging header name**—The name of the HTTP header that holds the Client IP, for logging.

- **Undefined profile**—The profile applied when the corresponding policy action evaluates as undefined.
- **Default profile**—The profile applied to connections that do not match a policy.
- **Import size limit**—The maximum cumulative total byte count of all files imported to the appliance, including signatures, WSDLs, schemas, HTML, and XML error pages. During an import, if the size of the imported object would cause the cumulative total sizes of all imported files to exceed the configured limit, the import operation fails and the appliance displays the following error message: *ERROR: Import failed - exceeding the configured total size limit on the imported objects*.
- **Learn message rate limit**—The maximum number of requests and responses per second that the learning engine is to process. Any additional requests or responses over this limit are not sent to the learning engine.
- **Entity decoding**—Decode HTML entities when running App Firewall checks.
- **Log malformed request**—Enable logging of malformed HTTP requests.
- **Use configurable secret key**—Use a configurable secret key for Web App Firewall operations. This secret key is used for signing and verifying data. When "useConfigurableSecretKey" is turned ON, you must use the key enabled in the "set ns encryptionParams" parameter.
- **Reset learned data**—Remove all learned data from the App Firewall. Restarts the learning process by collecting fresh data.

Two settings, *Reset Learned Data* and *Signatures Auto-Update*, are found in different places depending on whether you use the command line or the GUI to configure your App Firewall. When using the command line, you configure Reset Learned Data by using the reset appfw learning data command, which takes no parameters and has no other functions. You configure Signatures Auto-Update in the set appfw settings command: the -signatureAutoUpdate parameter enables or disables auto-updating of the signatures, and -signatureUrl configures the URL which hosts the updated signatures file.

When using the GUI, you configure Reset Learned Data in **Security > **App Firewall** > Engine Settings**; the **Reset Learned Data** button is at the bottom of the dialog box. You configure Signatures Auto-Update for each set of signatures in **Security** > **App Firewall** > **Signatures**, by selecting the signatures file, clicking the right mouse button and selecting **Auto Update Settings**.

Normally, the default values for the **App Firewall** settings are correct. If the default settings cause a conflict with other servers or cause premature disconnection of your users, however, you might need to modify them.

The **App Firewall** session limit is configurable using the following command:

```
> set appfw settings -sessionLimit 500000
Done
Default value:100000   Max value:500000 per PE
<!--NeedCopy-->
```

To configure engine settings by using the command line interface

At the command prompt, type the following commands:

- `set appfw settings [-sessionCookieName <name>] [-sessionTimeout <positiveInteger>] [-sessionLifetime <positiveInteger>][-clientIPLoggingHeader <headerName>] [-undefaction <profileName>] [-defaultProfile <profileName>] [-importSizeLimit <positiveInteger>] [-logMalformedReq (ON | OFF)] [-signatureAutoUpdate (ON | OFF)] [-signatureUrl <expression>] [-cookiePostEncryptPrefix <string>] [-entityDecoding (ON | OFF)] [-useConfigurableSecretKey (ON | OFF)][-learnRateLimit <positiveInteger>]`
- `save ns config`

Example

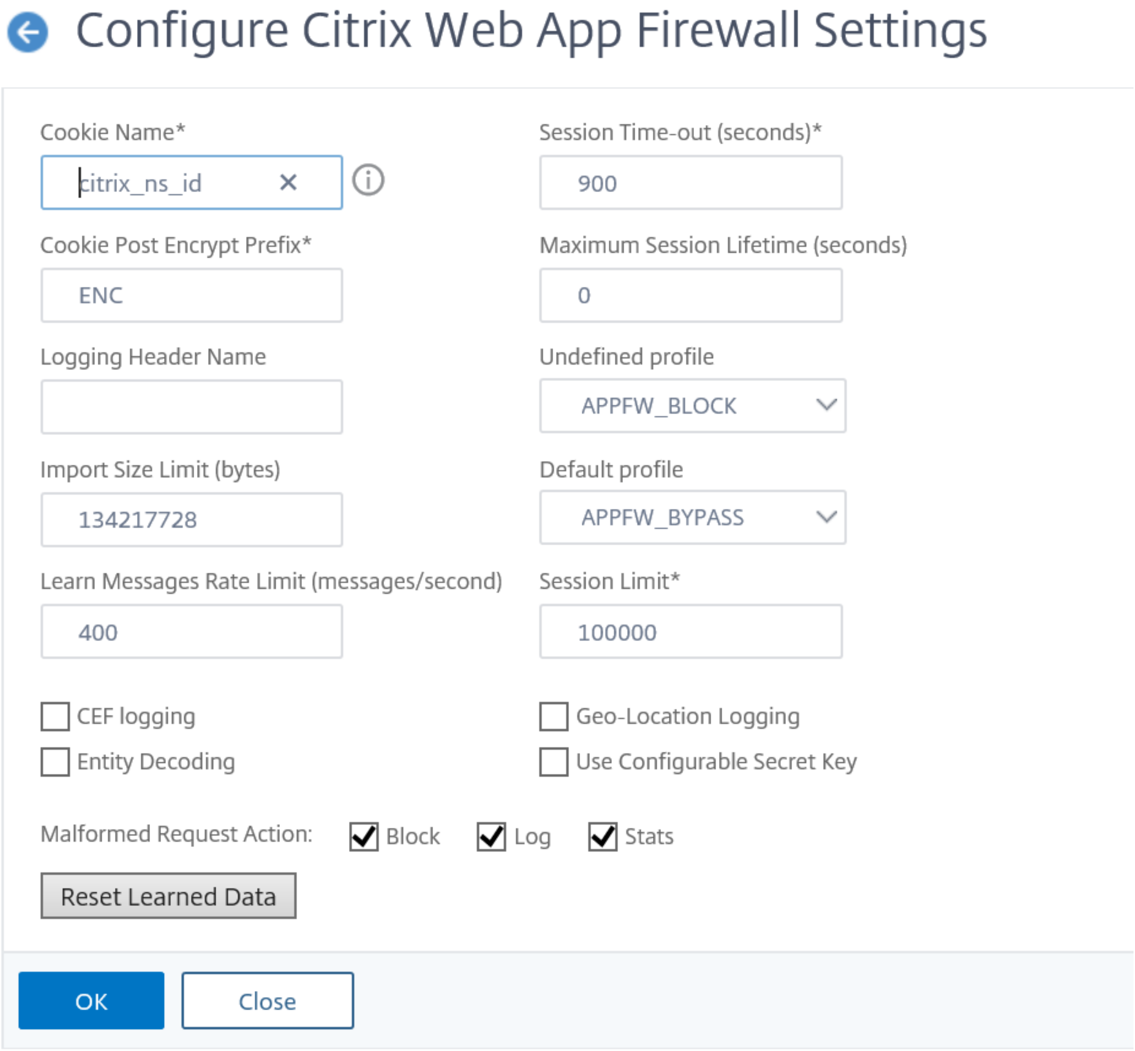
```
set appfw settings -sessionCookieName citrix-appfw-id -sessionTimeout 3600
-sessionLifetime 14400 -clientIPLoggingHeader NS-AppFW-Client-IP -undefaction APPFW_RESET
-defaultProfile APPFW_RESET -importSizeLimit 4096
save ns config
<!--NeedCopy-->
```

To configure engine settings by using the GUI

1. Navigate to **Security** > **App Firewall**
2. In the details pane, click **Change Engine Settings**.
3. In the **App Firewall Engine Settings** dialog box, set the following parameters:
    - Cookie Name
    - Session Timeout
    - Cookie Post Encrypt Prefix
    - Maximum Session Lifetime
    - Logging Header Name
    - Undefined Profile
    - Default Profile
    - Import Size Limit
    - Learn Messages Rate Limit
    - Entity Decoding
    - Log Malformed Request

- Use Secret Key
- Learn Message Rate Limit
- Signatures Auto Update

4. Click **OK**.



Configure Citrix Web App Firewall Settings

Cookie Name*: citrix_ns_id
Session Time-out (seconds)*: 900
Cookie Post Encrypt Prefix*: ENC
Maximum Session Lifetime (seconds): 0
Logging Header Name:
Undefined profile: APPFW_BLOCK
Import Size Limit (bytes): 134217728
Default profile: APPFW_BYPASS
Learn Messages Rate Limit (messages/second): 400
Session Limit*: 100000

☐ CEF logging
☐ Geo-Location Logging
☐ Entity Decoding
☐ Use Configurable Secret Key

Malformed Request Action: ☑ Block ☑ Log ☑ Stats

Confidential fields

September 21, 2020

You can designate web-form fields as confidential to protect the information users type into them. Normally, any information a user types into a web form on one of your protected web servers is logged in the NetScaler logs. The information typed into a web-form field designated as confidential, however, is not logged. That information is saved only where the web site is configured to save such data, normally in a secure database.

Common types of information that you may want to protect with a confidential field designation include:

- Passwords
- Credit card numbers, validation codes, and expiration dates
- Social security numbers
- Tax ID numbers
- Home addresses
- Private telephone numbers

In addition to being good practice, proper use of confidential field designations may be necessary for PCI-DSS compliance on ecommerce servers, HIPAA compliance on servers that manage medical information in the United States, and compliance with other data protection standards.

> **Important:**
>
> In the following two cases, the Confidential Field designation does not function as expected:

- If a Web form has either a confidential field or an action URL longer than 256 characters, the field or action URL is truncated in the NetScaler logs.
- With certain SSL transactions, the logs are truncated if either the confidential field or the action URL is longer than 127 characters.

In either of these cases, the App Firewall masks a fifteen-character string with the letter "x," instead of the normal eight character string. To ensure that any confidential information is removed, the user must use form field name and action URL expressions that match the first 256, or (in cases where SSL is used) the first 127 characters.

To configure your App Firewall to treat a web-form field on a protected web site as confidential, you add that field to the Confidential Fields list. You can enter the field name as a string, or you can enter a PCRE-compatible regular expression specifying one or more fields. You can enable the confidential-field designation when you add the field, or you can modify the designation later.

To add a confidential field by using the command line interface

At the command prompt, type the following commands:

- `add appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX )] [-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- `save ns config`

Example

The following example adds all web form fields whose names begin withPassword to the confidential fields list.

```
add appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

To modify a confidential field by using the command line interface

At the command prompt, type the following commands:

- `set appfw confidField <fieldName> <url> [-isRegex ( REGEX | NOTREGEX ) ][-comment "<string>"] [-state ( ENABLED | DISABLED )]`
- 'save ns config'

Example

The following example modifies the confidential field designation to add a comment.

```
set appfw confidField Password "https?://www[.]example[.]com/[^<>]*[^a-z]password[0-9a-z._-]*[.](asp|cgi|htm|html|htp|js|php)" -comment "Protect password fields." -isRegex REGEX -state ENABLED
save ns config
<!--NeedCopy-->
```

To remove a confidential field by using the command line interface

At the command prompt, type the following commands:

- `rm appfw confidField <fieldName> <url>`
- `save ns config`

To configure a confidential field by using the GUI

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage Confidential Fields**.
3. In the Manage Confidential Fields dialog box, do one of the following:
   - To add a new form field to the list, click Add.

  - To change an existing confidential field designation, select the field, and then click **Edit**. The **App Firewall Confidential Fields** dialog box appears.

    > **Note**:
    >
    > If you select an existing confidential field designation and then click **Add**, the **Create Confidential Form Field** dialog box displays the information for that confidential field. You can modify that information to create your new confidential field.

4. In the dialog box, fill out the elements. They are:
    - **Enabled check box.** Select or clear to enable/disable this confidential field designation.
    - **Is form field name a regular expression check box.** Select or clear to enable PCRE-format regular expressions in the form field name.
    - **Field Name.** Enter a literal string or PCRE-format regular expression that either represents a specific field name or that matches multiple fields with names that follow a pattern.
    - **Action URL.** Enter a literal URL or a regular expression that defines one or more URLs of the web page(s) on which the web form(s) that contains the confidential field are located.
    - **Comments.** Enter a comment. Optional.
5. Click **Create** or **OK**.
6. To remove a confidential field designation from the confidential fields list, select the confidential field listing you want to remove, then click Remove to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing confidential field designations, click **Close**.

Examples

Following are some regular expressions that define form field names that you might find useful:

- ^passwd_ (Applies confidential-field status to all field names that begin with the "passwd_" string.)

^(([0-9a-zA-Z._-]*	\x[0-9A-Fa-f][0-9A-Fa-f])+-)?passwd_ (Applies confidential-field status to all field names that begin with the string passwd_, or that contain the string -passwd_ after another string that might contain non-ASCII special characters.)

-

Following are some regular expressions that define specific URL types that you might find useful. Substitute your own web host(s) and domain(s) for those in the examples.

- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you could use the following regular expression:

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)*logon[.]pl\?
<!--NeedCopy-->
```

- If the web form appears on multiple web pages on the web host www.example-español.com, which contains the n-tilde (ñ) special character, you could use the following regular expression, which represents the n-tilde special character as an encoded UTF-8 string containing C3 B1, the hexadecimal code assigned to that character in the UTF-8 charset:

```
https?://www[.]example-espa\xC3\xB1ol[.]com/([0-9A-Za-z][0-9A-Za-z_.-]*/)* logon[.]pl\?
<!--NeedCopy-->
```

- If the web form containing query.pl appears on multiple web pages on different hosts within the example.com domain, you could use the following regular expression:

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

- If the web form containing query.pl appears on multiple web pages on different hosts in different domains, you could use the following regular expression:

```
https?://([0-9A-Za-z][0-9A-Za-z_-.]*[.])*[0-9A-Za-z][0-9A-Za-z_-.]+[.][a-z]{
2,6 }
/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

- If the web form appears on multiple web pages on the web host www.example.com, but all of those web pages are named logon.pl?, you could use the following regular expression:

```
https?://www[.]example[.]com/([0-9A-Za-z][0-9A-Za-z_-.]*/)*logon[.]pl\?
<!--NeedCopy-->
```

Field types

September 21, 2020

A field type is a PCRE-format regular expression that defines a particular data format and minimum/-maximum data lengths for a form field in a web form. Field types are used in the Field Formats check.

The App Firewall comes with several default field types, which are:

- integer. A string of any length consisting of numbers only, without a decimal point, and with an optional preceding minus sign (-).
- alpha. A string of any length consisting of letters only.
- alphanum. A string of any length consisting of letters and/or numbers.
- nohtml. A string of any length consisting of characters, including punctuation and spaces, that does not contain HTML symbols or queries.
- any. Anything at all.

  > **Important**:
  >
  > Assigning the any field type as the default field type, or to a field, allows active scripts, SQL commands, and other possibly dangerous content to be sent to your protected web sites and applications in that form field. You should use the any type sparingly, if you use it at all.

You can also add your own field types to the Field Types list. For example, you might want to add a field type for a social security number, postal code, or phone number in your country. You might also want to add a field type for a customer identification number or store credit card number.

To add a field type to the Field Types list, you enter the field name as a literal string or PCRE-format regular expression.

To add a field type by using the command line interface

At the command prompt, type the following commands:

- `add appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example adds a field type named SSN that matches US Social Security numbers to the Field Types list, and sets its priority to 1.

```
add appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1
save ns config
<!--NeedCopy-->
```

To modify a field type by using the command line interface

At the command prompt, type the following commands:

- `set appfw fieldType <name> <regex> <priority> [-comment "<string>"]`
- `save ns config`

Example

The following example modifies the field type to add a comment.

```
set appfw fieldType SSN "^[1-9][0-9]{
 2,2 }
 -[0-9 }
 {
 2,2 }
 -[0-9]{
 4,4 }
 $" 1 -comment "US Social Security Number"
save ns config
<!--NeedCopy-->
```

To remove a field type by using the command line interface

At the command prompt, type the following commands:

- `>rm appfw fieldType <name>`
- `save ns config`

To configure a field type by using the GUI

1. Navigate to Security > Application Firewall.
2. In the details pane, under **Settings**, click **Manage Field Types**.
3. In the **Manage Field Types** dialog box, do one of the following:
    - To add a new field type to the list, click **Add**.
    - To change an existing field type, select the field type, and then click **Edit**.

    The **Configure Field Type** dialog box appears.

    > **Note**:
    >
    > If you select an existing field type designation and then click **Add**, the dialog box displays the information for that field type. You can modify that information to create your new field type.
4. In the dialog box, fill out the elements. They are:
    - Name
    - Regular Expression
    - Priority
    - Comment
5. Click Create or OK.
6. To remove a field type from the Field Types list, select the field type listing you want to remove, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding, modifying, and removing field types, click **Close**.

Examples

Following are some regular expressions for field types that you might find useful:

`^[1-9][0-9]{ 2,2 } -[0-9 } { 2,2 } -[0-9]{ 4,4 } $ U.S. Social Security numbers`

`^\[A-C\]\[0-9\]{ 7,7 } $ California driver's license numbers`

`^+[0-9]{ 1,3 } [0-9()-]{ 1,40 } $ International phone numbers with country codes`

`^[0-9]{ 5,5 } -[0-9]{ 4,4 } $ U.S. ZIP code numbers`.

`^[0-9A-Za-z][0-9A-Za-z.+_-]{ 0,25 } @([0-9A-Za-z][0-9A-Za-z_-]*[.]){ 1,4 } [A-Za-z]{ 2,6 } $ Email addresses`‘.

XML content types

September 21, 2020

By default, the App Firewall treats files that follow certain naming conventions as XML. You can configure the App Firewall to examine web content for additional strings or patterns that indicate that those files are XML files. This can ensure that the App Firewall recognizes all XML content on your site, even if certain XML content does not follow normal XML naming conventions, ensuring that XML content is subjected to XML security checks.

To configure the XML content types, you add the appropriate patterns to the XML Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing XML content types patterns.

To add an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- `add appfw XMLContentType <XMLContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

Example

The following example adds the pattern .*/xml to the XML Content Types list and designates it as a regular expression.

```
add appfw XMLContentType ".*/xml" -isRegex REGEX
<!--NeedCopy-->
```

To remove an XML content type pattern by using the command line interface

At the command prompt, type the following commands:

- `rm appfw XMLContentType <XMLContenttypevalue>`
- `save ns config`

To configure the XML content type list by using the GUI

1. Navigate to **Security** > **App Firewall**.
2. In the details pane, under **Settings**, click **Manage XML Content Types**.
3. In the **Manage XML Content Types** dialog box, do one of the following:
    - To add a new XML content type, click Add.
    - To modify an existing XML content type, select that type and then click Edit.
      The Configure App Firewall XML Content Type dialog appears. Note: If you select an existing XML content type pattern and then click

Add, the dialog box displays the information for that XML content type pattern. You can modify that information to create your new XML content type pattern.

4. In the dialog box, fill out the elements. They are:
    - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
    - **XML Content Type** Enter a literal string or PCRE-format regular expression that matches the XML content type pattern that you want to add.
5. Click **Create**.
6. To remove an XML content type pattern from the list, select it, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click **Close**.

JSON content types

September 21, 2020

By default, the App Firewall treats files with the content type "application/json" as JSON files.The default setting enables the App Firewall to recognize JSON content in requests and responses, and to handle that content appropriately.

You can configure the App Firewall to examine web content for additional strings or patterns that indicate that those files are JSON files. This can ensure that the App Firewall recognizes all JSON content on your site, even if certain JSON content does not follow normal JSON naming conventions, ensuring that JSON content is subjected to JSON security checks.

To configure the JSON content types, you add the appropriate patterns to the JSON Content Types list. You can enter a content type as a string, or you can enter a PCRE-compatible regular expression specifying one or more strings. You can also modify the existing JSON content types patterns.

To add a JSON content type pattern by using the command line interface

At the command prompt, type the following commands:

- `add appfw JSONContentType <JSONContenttypevalue> [-isRegex ( REGEX | NOTREGEX )]`
- `save ns config`

Example

The following example adds the pattern .*/json to the JSON Content Types list and designates it as a regular expression.

```
add appfw JSONContentType ".*/json" -isRegex REGEX
<!--NeedCopy-->
```

To configure the JSON content type list by using the GUI

1. Navigate to **Security** > **Application Firewall**.
2. In the details pane, under **Settings**, click **Manage JSON Content Types**.
3. In the Manage JSON Content Types dialog box, do one of the following:
    - To add a new JSON content type, click Add.
    - To modify an existing JSON content type, select that type and then click Edit.
      The Configure App Firewall JSON Content Type dialog appears.
      Note: If you select an existing JSON content type pattern and then click
      Add, the dialog box displays the information for that JSON content type pattern. You can modify that information to create your new JSON content type pattern.
4. In the dialog box, fill out the elements. They are:
    - **IsRegex.** Select or clear to enable PCRE-format regular expressions in the form field name.
    - **JSON Content Type** Enter a literal string or PCRE-format regular expression that matches the JSON content type pattern that you want to add.
5. Click **Create** or **OK**.
6. To remove a JSON content type pattern from the list, select it, then click **Remove** to remove it, and then click **OK** to confirm your choice.
7. When you have finished adding and removing XML content type patterns, click **Close**.

Statistics and reports

September 21, 2020

The information maintained in the logs and statistics, and displayed in the reports, provides important guidance for configuring and maintaining the App Firewall.

The App Firewall statistics

When you enable the statistics action for App Firewall signatures or security checks, the App Firewall maintains information about connections that match that signature or security check. You can view the accumulated statistics information on the
Monitoring tab of the main logon page of your App Firewall appliance by selecting one of the following choices in the Select Group list box:

- **App Firewall.** A summary of all statistics information gathered by your App Firewall appliance for all profiles.
- **App Firewall (per profile).** The same information, but displayed per-profile rather than summarized.

You can use this information to monitor how your App Firewall is operating and determine whether there is any abnormal activity or abnormal amounts of hits on a signature or security check. If you see such a pattern of abnormal activity, you can check the logs for that signature or security check, to diagnose the issue, and then take corrective action.

The App Firewall reports

The App Firewall reports provide information about your App Firewall configuration and how it is handling traffic for your protected web sites.

The PCI DSS report

The Payment Card Industry (PCI) Data Security Standard (DSS), version 1.2, consists of twelve security criteria that most credit card companies require businesses who accept online payments via credit and debit cards to meet. These criteria are designed to prevent identity theft, hacking, and other types of fraud. If an internet service provider or online merchant does not meet the PCI DSS criteria, that ISP or merchant risks losing authorization to accept credit card payments through its web site.

ISPs and online merchants prove that they are in compliance with PCI DSS by having an audit conducted by a PCI DSS Qualified Security Assessor (QSA) Company. The PCI DSS report is designed to assist them both before and during the audit. Before the audit, it shows which App Firewall settings are relevant to PCI DSS, how they should be configured, and (most important) whether your current App Firewall configuration meets the standard. During the audit, the report can be used to demonstrate compliance with relevant PCI DSS criteria.

The PCI DSS report consists of a list of those criteria that are relevant to your App Firewall configuration. Under each criterion, it lists your current configuration options, indicates whether your current configuration complies with the PCI DSS criterion, and explains how to configure the App Firewall so that your protected web site(s) will be in compliance with that criterion.

The PCI DSS report is located under **System** > **Reports**. To generate the report as an Adobe PDF file, click **Generate PCI DSS Report**. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

> **Note**:
> To view this and other reports, you must have the Adobe Reader program installed on your computer.

The PCI DSS report consists of the following sections:

- **Description.** A description of the PCI DSS Compliance Summary report.
- **Firewall License and Feature Status.** Tells you whether the App Firewall is licensed and enabled on your NetScaler appliance.
- **Executive Summary.** A table that lists the PCI DSS criteria and tells you which of those criteria are relevant to the App Firewall.
- **Detailed PCI DSS Criteria Information.** For each PCI DSS criterion that is relevant to your App Firewall configuration, the PCI DSS report provides a section that contains information about whether your configuration is currently in compliance and, if it is not, how to bring it into compliance.
- **Configuration.** Data for individual profiles, which you access either by clicking App Firewall Configuration at the top of the report, or directly from the Reports pane. The App Firewall Configuration report is the same as the PCI DSS report, with the PCI DSS-specific summary omitted, and is described below.

The App Firewall configuration report

The App Firewall Configuration report is located under System > Reports. To display it, click Generate App Firewall Configuration Report. Depending on your browser settings, the report is displayed in the pop-up window or you are prompted to save it to your hard disk.

The App Firewall Configuration report starts with a Summary page, which consists of the following sections:

- **App Firewall Policies.** A table that lists your current App Firewall policies, showing the policy name, the content of the policy, the action (or profile) it is associated with, and global binding information.
- **App Firewall Profiles.** A table that lists your current App Firewall profiles and indicates which policy each profile is associated with. If a profile is not associated with a policy, the table displays INACTIVE in that location.

To download all report pages for all policies, at the top of the Profiles Summary page click Download All Profiles. You display the report page for each individual profile by selecting that profile in the table at the bottom of the screen. The Profile page for an individual profile shows whether each check action is enabled or disabled for each check, and the other configuration settings for the check.

To download a PDF file containing the PCI DSS report page for the current profile, click **Download Current Profile** at the top of the page. To return to the Profiles Summary page, click **App Firewall Profiles**. To go back to the main page, click **Home**. You can refresh the PCI DSS report at any time by clicking **Refresh** in the upper right corner of the browser. You should refresh the report if you make changes to your configuration.

App Firewall logs

October 28, 2020

The App Firewall generated log messages can be quite useful for keeping track of the configurational changes, App Firewall policy invocations, and security check violations.

When the log action is enabled for security checks or signatures, the resulting log messages provide information about the requests and responses that the App Firewall has observed while protecting your web sites and applications. The most important information is the action taken by the App Firewall when a signature or a security check violation was observed. For some security checks, the log message can provide additional useful information, such as the location and the detected pattern that triggered the violation. You can deploy security checks in non-block mode and monitor the logs to determine whether the transactions that are triggering security violations are valid transactions (false positives). If they are, you can either remove, or reconfigure the signature or security checks, deploy relaxations, or take other appropriate measures to mitigate the false positives before you enable blocking for that signature or security check. An excessive increase in the number of violation messages in logs can indicate a surge in malicious requests. This can alert you that your application might be under attack to exploit a specific vulnerability that is detected and thwarted by App Firewall protections.

> **Note:**
>
> Citrix Web App Firewall logging must be used only with external SYSLOG servers.

NetScaler (Native) format logs

The App Firewall uses the NetScaler format logs (also called native format logs) by default. These logs have the same format as those generated by other NetScaler features. Each log contains the following fields:

- Timestamp. Date and time when the connection occurred.
- Severity. Severity level of the log.
- Module. NetScaler module that generated the log entry.
- Event Type. Type of event, such as signature violation or security check violation.
- Event ID. ID assigned to the event.
- Client IP. IP address of the user whose connection was logged.
- Transaction ID. ID assigned to the transaction that caused the log.
- Session ID. ID assigned to the user session that caused the log.
- Message. The log message. Contains information identifying the signature or security check that triggered the log entry.

You can search for any of these fields, or any combination of information from different fields. Your selection is limited only by the capabilities of the tools you use to view the logs. You can observe the App Firewall log messages in the GUI by accessing the NetScaler syslog viewer, or you can manually connect to the NetScaler appliance and access logs from the command line interface, or you can drop into shell and tail the logs directly from the /var/log/ folder.

Example of a Native Format Log message

```
Jun 22 19:14:37 <local0.info> 10.217.31.98 06/22/2015:19:14:37 GMT ns
    0-PPE-1 :
default APPFW APPFW_XSS 60 0 :  10.217.253.62 616-PPE1 y/3
    upt2K8ySWWId3Kavbxyni7Rw0000
pr_ffc http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
12345&drinking_pref=on&text_area=%3Cscript%3E%0D%0A&loginButton=
    ClickToLogin&as_sfid=
AAAAAAWEXcNQLlSokNmqaYF6dvfqlChNzSMsdyO9JXOJomm2v
BwAMOqZIChv21EcgBc3rexIUcfm0vckKlsgoOeC_BArx1Ic4NLxxkWMtrJe4H7SOfkiv9NL7AG4juPIan

%3D&as_fid=feeec8758b41740eedeeb6b35b85dfd3d5def30c Cross-site script
    check failed for
field text_area="Bad tag: script" <blocked>
<!--NeedCopy-->
```

Common Event Format (CEF) Logs

The App Firewall also supports CEF logs. CEF is an open log management standard that improves the interoperability of security-related information from different security and network devices and applications. CEF enables customers to use a common event log format so that data can easily be collected and aggregated for analysis by an enterprise management system. The log message is broken into different fields so that you can easily parse the message and write scripts to identify important information.

Analyzing the CEF Log Message

In addition to date, timestamp, client IP, log format, appliance, company, build version, module, and security check information, App Firewall CEF Log messages include the following details:

- src – source IP address
- spt – source port number
- request – request URL
- act – action (e.g. blocked, transformed)
- msg – message (Message regarding the observed security check violation)
- cn1 – event ID
- cn2 – HTTP Transaction ID

- cs1 – profile name
- cs2 – PPE ID (e.g. PPE1)
- cs3 - Session ID
- cs4 – Severity (e.g. INFO, ALERT)
- cs5 – event year
- cs6 - Signature violation category
- method – Method (e.g. GET/POST)

For example, consider the following CEF format log message, which was generated when a Start URL violation was triggered:

```
Jun 12 23:37:17 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0
|APPFW|APPFW_STARTURL|6|src=10.217.253.62 spt=47606 method=GET
request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal URL. cn1=1340
cn2=653 cs1=pr_ffc cs2=PPE1 cs3=EsdGd3VD0OaaURLcZnj05Y6DOmE0002 cs4=ALERT cs5=2015
act=blocked
<!--NeedCopy-->
```

The above message can be broken down in the following components:

The above message can be broken down into different components. Refer to the CEP log components table.

Example of a request check violation in CEF log format: request is not blocked

```
Jun 13 00:21:28 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_FIELDCONSISTENCY|6|src=10.217.253.62 spt=761 method=GET request=
http://aaron.stratum8.net/FFC/login.php?login_name=abc&passwd=
123456789234&drinking_pref=on&text_area=&loginButton=ClickToLogin&as_sfid
=AAAAAAWIahZuYoIFbjBhYMP05mJLTwEfIY0a7AKGMg3jIBaKmwtK4t7M7lNxOgj7Gmd3SZc8KUj6CF
7W5kIWDRHN8PtK1Zc-txHkHNx1WknuG9DzTuM7t1THhluevXu9I4kp8%3D&as_fid=feeec8758b4174
0eedeeb6b35b85dfd3d5def30c msg=Field consistency check failed for field passwd cn1=1401
cn2=707 cs1=pr_ffc cs2=PPE1 cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=
not blocked
<!--NeedCopy-->
```

Example of a response check violation in CEF format: response is transformed

```
Jun 13 00:25:31 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SAFECOMMERCE|6|src=10.217.253.62 spt=34041 method=GET request=
http://aaron.stratum8.net/FFC/CreditCardMind.html msg=Maximum number of
     potential credit
card numbers seen cn1=1470 cn2=708 cs1=pr_ffc cs2=PPE1
cs3=Ycby5IvjL6FoVa6Ah94QFTIUpC80001 cs4=ALERT cs5=2015 act=transformed
<!--NeedCopy-->
```

Example of a request side signature violation in CEF format: request is blocked

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW|
APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method=GET request=
http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=Signature
    violation rule ID 807:
web-cgi /wwwboard/passwd.txt access  cn1=140 cn2=841 cs1=pr_ffc cs2=
    PPE0
cs3=OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
    blocked
<!--NeedCopy-->
```

Logging geolocation in the App Firewall violation messages

Geolocation, which identifies the geographic location from which requests originate, can help you configure the App Firewall for the optimal level of security. To bypass security implementations such as rate limiting, which rely on the IP addresses of the clients, malware or rogue computers can keep changing the source IP address in requests. Identifying the specific region from where requests are coming can help determine whether the requests are from a valid user or a device attempting to launch cyberattacks. For example, if an excessively large number of requests are received from a specific area, it is easy to determine whether they are being sent by users or a rogue machine. Geolocation analysis of the received traffic can be very useful in deflecting attacks such as denial of service (DoS) attacks.

The App Firewall offers you the convenience of using the built-in NetScaler database for identifying the locations corresponding to the IP addresses from which malicious requests are originating. You can then enforce a higher level of security for requests from those locations. Citrix default syntax (PI) expressions give you the flexibility to configure location based policies that can be used in conjunction with the built-in location database to customize firewall protection, bolstering your defense against coordinated attacks launched from rogue clients in a specific region.

You can use the NetScaler built-in database, or you can use any other database. If the database does not have any location information for the particular client IP address, the CEF log shows geolocation as an Unknown geolocation.

Note: Geolocation logging uses the Common Event Format (CEF). By default, CEF logging and GeoLocationLogging are OFF. You must explicitly enable both parameters.

Example of a CEF log message showing geolocation information

```
June 8 00:21:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|APPFW|
APPFW_STARTURL|6|src=10.217.253.62 geolocation=NorthAmerica.US.Arizona.Tucson.*.*
spt=18655 method=GET request=http://aaron.stratum8.net/FFC/login.html
msg=Disallow Illegal URL. cn1=77 cn2=1547 cs1=test_pr_adv cs2=PPE1
cs3=KDynjg1pbFtfhC/nt0rBU1o/Tyg0001 cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

Example of a log message showing geolocation= Unknown

```
June 9 23:50:53 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11.0|
APPFW|APPFW_STARTURL|6|src=10.217.30.251 geolocation=Unknown spt=5086
method=GET request=http://aaron.stratum8.net/FFC/login.html msg=Disallow Illegal URL.
cn1=74 cn2=1576 cs1=test_pr_adv cs2=PPE2 cs3=PyR0eOEM4gf6GJiTyauiHByL88E0002
cs4=ALERT cs5=2015 act=not blocked
<!--NeedCopy-->
```

Using the command line to configure the log action and other log parameters

To configure the log action for a security checks of a profile by using the command line

At the command prompt, type one of the following commands:

- `set appfw profile <name> SecurityCheckAction ([log] | [none])`
- `unset appfw profile <name> SecurityCheckAction`

Examples

`set appfw profile pr_ffc StartURLAction log`

`unset appfw profile pr_ffc StartURLAction`

To configure CEF logging by using the command line

The CEF logging is disabled by default. At the command prompt, type one of the following commands to change or display the current setting:

- `set appfw settings CEFLogging on`
- `unset appfw settings CEFLogging`
- `sh appfw settings | grep CEFLogging`

To configure the logging of the credit card numbers by using the command line

At the command prompt, type one of the following commands:

- `set appfw profile <name> -doSecureCreditCardLogging ([ON] | [OFF])`
- `unset appfw profile <name> -doSecureCreditCardLogging`

To configure Geolocation logging by using the command line

1. Use the set command to enable GeoLocationLogging. You can enable the CEF logging at the same time. Use the unset command to disable geolocation logging. The show command shows the current settings of all the App Firewall parameters, unless you include the grep command to show the setting for a specific parameter.
    - `set appfw settings GeoLocationLogging ON [CEFLogging ON]`
    - `unset appfw settings GeoLocationLogging`
    - `sh appfw settings | grep GeoLocationLogging`
2. Specify the database

    `add locationfile /var/netscaler/inbuilt_db/Citrix_netscaler_InBuilt_GeoIP_DB.csv`

    or

    `add locationfile <path to database file>`

Customizing App Firewall Logs

Default format (PI) expressions give you the flexibility to customize the information included in the logs. You have the option to include the specific data that you want to capture in the App Firewall generated log messages. For example, if you are using AAA-TM authentication along with the App Firewall security checks, and would like to know the accessed URL that triggered the security check violation, the name of the user who requested the URL, the source IP address, and the source port from which the user sent the request, you can use the following commands to specify customized log messages that include all the data:

```
> sh version
NetScaler NS12.1: Build 50.0013.nc, Date: Aug 28 2018, 10:51:08   (64-bit)
 Done
```

```
<!--NeedCopy-->
```

```
> add audit messageaction custom1 ALERT 'HTTP.REQ.URL + " " + HTTP.REQ.
   USER.NAME + " " + CLIENT.IP.SRC + ":" + CLIENT.TCP.SRCPORT'
Warning: HTTP.REQ.USER has been deprecated. Use AAA.USER instead.
 Done
<!--NeedCopy-->
```

```
> add appfw profile test_profile
 Done
<!--NeedCopy-->
```

```
> add appfw policy appfw_pol true test_profile -logAction custom1
 Done
<!--NeedCopy-->
```

Configuring Syslog policy to segregate App Firewall logs

The App Firewall offers you an option to isolate and redirect the App Firewall security log messages to a different log file. This might be desirable if the App Firewall is generating a large number of logs, making it difficult to view other NetScaler log messages. You can also use this option when you are interested only in viewing the App Firewall log messages and do not want to see the other log messages.

To redirect the App Firewall logs to a different log file, configure a syslog action to send the App Firewall logs to a different log facility. You can use this action when configuring the syslog policy, and bind it globally for use by App Firewall.

Example:

1. Switch to the shell and use an editor such as vi to edit the /etc/syslog.conf file. Add a new entry to use local2.* to send logs to a separate file as shown in the following example:

   `local2.\* /var/log/ns.log.appfw`

2. Restart the syslog process. You can use the grep command to identify the syslog process ID (PID), as shown in the following example:

   `root@ns\## ps -A | grep syslog`

   `1063 ?? Ss 0:03.00 /usr/sbin/syslogd -b 127.0.0.1 -n -v -v -8 -C`

   `root@ns## kill -HUP 1063`

3. From the command line interface, configure the syslog action and policy. Bind it as a global App Firewall policy.

```
> add audit syslogAction sysact 1.1.1.1 -logLevel ALL -logFacility LOCAL2
```

```
> add audit syslogPolicy syspol1 ns_true sysact1
```

```
> bind appfw global syspol1 100
```

1. All App Firewall security check violations will now be redirected to the /var/log/ns.log.appfw file. You can tail this file to view the App Firewall violations that are getting triggered during the processing of the ongoing traffic.

   ```
   root@ns## tail -f ns.log.appfw
   ```

Warning: If you have configured the syslog policy to redirect the logs to a different log facility, the App Firewall log messages no longer appear in the /var/log/ns.log file.

Viewing the App Firewall Logs

You can view the logs by using the syslog viewer, or by logging onto the NetScaler appliance, opening a UNIX shell, and using the UNIX text editor of your choice.

To access the log messages by using the command line

Switch to the shell and tail the ns.logs in the /var/log/ folder to access the log messages pertaining to the App Firewall security check violations:

- `Shell`
- `tail -f /var/log/ns.log`

You can use the vi editor, or any Unix text editor or text search tool, to view and filter the logs for specific entries. For example, you can use grep command to access the log messages pertaining to the Credit Card violations:

- `tail -f /var/log/ns.log | grep SAFECOMMERCE`

To access the log messages by using the GUI

The NetScaler GUI includes a very useful tool (Syslog Viewer) for analyzing the log messages. You have multiple options for accessing the Syslog Viewer:

- To view log messages for a specific security check of a profile, navigate to **App Firewall** > **Profiles**, select the target profile, and click Security Checks. Highlight the row for the target security check and click Logs. When you access the logs directly from the selected security check of the profile, it filters out the log messages and displays only the logs pertaining to the violations for the selected security check. Syslog viewer can display App Firewall logs in the Native format as well as the CEF format. However, in order for the syslog viewer to filter out the target profile specific log messages, the logs must be in the CEF log format when accessed from the profile.
- You can also access the Syslog Viewer by navigating to **NetScaler** > **System** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays

all log messages, including all App Firewall security check violation logs for all profiles. This is useful for debugging when multiple security check violations might be triggered during request processing.
- Navigate to **App Firewall** > **policies** > **Auditing**. In the Audit Messages section, click Syslog messages link to display the Syslog Viewer, which displays all log messages, including all security check violation logs for all profiles.

The HTML based Syslog Viewer provides the following filter options for selecting only the log messages that are of interest to you:

- **File**—The current `/var/log/ns.log` file is selected by default, and the corresponding messages appear in the Syslog Viewer. A list of other log files in the /var/log directory are available in a compressed .gz format. To download and un-compress an archived log file, just select the log file from the dropdown option. The log messages pertaining to the selected file are then displayed in the syslog viewer. To refresh the display, click the Refresh icon (a circle of two arrows).

- **Module list box**—You can select the NetScaler module whose logs you want to view. You can set it to APPFW for App Firewall logs.

- **Event Type list box**—This box contains a set of check boxes for selecting the type of event you are interested in. For example, to view the log messages pertaining to the signature violations, you can select the **APPFW_SIGNATURE_MATCH** check box. Similarly, you can select a check box to enable the specific security check that is of interest to you. You can select multiple options.

- **Severity**—You can select a specific severity level to show just the logs for that severity level. Leave all the check boxes blank if you want to see all logs.

  To access the App Firewall security check violation log messages for a specific security check, filter by selecting **APPFW** in the dropdown options for Module. The Event Type displays a rich set of options to further refine your selection. For example, if you select the **APPFW_FIELDFORMAT** check box and click the Apply button, only log messages pertaining to the Field Formats security check violations appear in the Syslog Viewer. Similarly, if you select the **APPFW_SQL** and **APPFW_STARTURL** check boxes and click the **Apply** button, only log messages pertaining to these two security check violations will appear in the syslog viewer.

If you place the cursor in the row for a specific log message, multiple options, such as **Module**, **EventType**, **EventID**, **ClientIP**, **TransactionID**, and so on appear below the log message. You can select any of these options to highlight the corresponding information in the logs.

Click to Deploy: This functionality is available only in the GUI. You can use the Syslog Viewer to not only view the logs but also to deploy relaxation rules based on the log messages for the App Firewall security check violations. The log messages must be in CEF log format for this operation. If the relaxation rule can be deployed for a log message, a check box appears at the right edge of the Syslog Viewer box in the row. Select the check box, and then select an option from the Action list to deploy

the relaxation rule.
Edit & Deploy,
Deploy, and
Deploy All are available as Action options. For example, you can select an individual log message to edit and deploy. You can also select the check boxes for multiple log messages from one or more security checks and use the Deploy or Deploy All option. Click to Deploy functionality is currently supported for the following security checks:

- StartURL
- URL Buffer overflow
- SQL Injection
- XSS
- Field consistency
- Cookie consistency

To use Click to Deploy functionality in the GUI

1. In the **Syslog Viewer**, select **APPFW** in the **Module** options.
2. Select the security check for which to filter corresponding log messages.
3. Enable the check box to select the rule.
4. Use the **Action** drop-down list of options to deploy the relaxation rule.
5. Verify that the rule appears in the corresponding relaxation rule section.

> **Note**:
> SQL Injection and XSS rules that are deployed by using Click **Deploy** option do not include the fine grain relaxation recommendations.

Highlights

- **CEF Log Format support**—The CEF log format option provides a convenient option to monitor, parse, and analyze the App Firewall log messages to identify attacks, fine tune configured settings to decrease false positives, and gather statistics.
- **Click to Deploy**—The Syslog viewer provides an option to filter, evaluate, and deploy relaxation rules for single or multiple security check violations from one convenient location.
- **Option to customize log message**—You can use advanced PI expressions to customize log messages and include the data you want to see in the logs.
- **Segregate App Firewall specific logs**—You have an option to filter and redirect application-firewall specific logs to a separate log file.
- **Remote Logging**—You can redirect the log messages to a remote syslog server.
- **Geolocation Logging**—You can configure the App Firewall to include the geolocation of the area from where the request is received. A built-in geolocation database is available, but you have the option to use an external geolocation database. The NetScaler appliance supports both IPv4

and IPv6 static geolocation databases.

- **Information rich log message**—Following are some examples of the type of information that can be included in the logs, depending on the configuration:
    - An App Firewall policy was triggered.
    - A security check violation was triggered.
    - A request was considered to be malformed.
    - A request or the response was blocked or not blocked.
    - Request data (such as SQL or XSS special characters) or response data (such as Credit card numbers or safe object strings) was transformed.
    - The number of credit cards in the response exceeded the configured limit.
    - The credit card number and type.
    - The log strings configured in the signature rules, and the signature ID.
    - Geolocation information about the source of the request.
    - Masked (X'd out) user input for protected confidential fields.

Appendices

September 21, 2020

The following supplemental material provides additional detail about complex or peripheral App Firewall tasks.

PCRE character encoding format

September 21, 2020

The NetScaler operating system supports direct entry of characters in the printable ASCII character set only—characters with hexadecimal codes between HEX 20 (ASCII 32) and HEX 7E (ASCII 127). To include a character with a code outside that range in your App Firewall configuration, you must enter its UTF-8 hexadecimal code as a PCRE regular expression.

A number of character types require encoding using a PCRE regular expression if you include them in your App Firewall configuration as a URL, form field name, or Safe Object expression. They include:

- **Upper-ASCII characters.** Characters with encodings from HEX 7F (ASCII 128) to HEX FF (ASCII 255). Depending on the character map used, these encodings can refer to control codes, ASCII characters with accents or other modifications, non-Latin alphabet characters, and symbols not included in the basic ASCII set. These characters can appear in URLs, form field names, and safe object expressions.

- **Double-Byte characters.** Characters with encodings that use two 8-byte words. Double-byte characters are used primarily for representing Chinese, Japanese, and Korean text in electronic format. These characters can appear in URLs, form field names, and safe object expressions.

  ASCII control characters. Non-printable characters used to send commands to a printer. All ASCII characters with hexadecimal codes less than HEX 20 (ASCII 32) fall into this category. These characters should never appear in a URL or form field name, however, and would rarely if ever appear in a safe object expression.

The NetScaler appliance does not support the entire UTF-8 character set, but only the characters found in the following eight charsets:

- **English US (ISO-8859-1).** Although the label reads, "English US," the App Firewall supports all characters in the ISO-8859-1 character set, also called the Latin-1 character set. This character set fully represents most modern western European languages and represents all but a few uncommon characters in the rest.
- **Chinese Traditional (Big5).** The App Firewall supports all characters in the BIG5 character set, which includes all of the Traditional Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in Hong Kong, Macau, Taiwan, and by many people of Chinese ethnic heritage who live outside of mainland China.
- **Chinese Simplified (GB2312).** The App Firewall supports all characters in the GB2312 character set, which includes all of the Simplified Chinese characters (ideographs) commonly used in modern Chinese as spoken and written in mainland China.
- **Japanese (SJIS).** The App Firewall supports all characters in the Shift-JIS (SJIS) character set, which includes most characters (ideographs) commonly used in modern Japanese.
- **Japanese (EUC-JP).** The App Firewall supports all characters in the EUC-JP character set, which includes all characters (ideographs) commonly used in modern Japanese.
- **Korean (EUC-KR).** The App Firewall supports all characters in the EUC-KR character set, which includes all characters (ideographs) commonly used in modern Korean.
- **Turkish (ISO-8859-9).** The App Firewall supports all characters in the ISO-8859-9 character set, which includes all letters used in modern Turkish.
- **Unicode (UTF-8).** The App Firewall supports certain additional characters in the UTF-8 character set, including those used in modern Russian.

When configuring the App Firewall, you enter all non-ASCII characters as PCRE-format regular expressions using the hexadecimal code assigned to that character in the UTF-8 specification. Symbols and characters within the normal ASCII character set, which are assigned single, two-digit codes in that character set, are assigned the same codes in the UTF-8 character set. For example, the exclamation point (!), which is assigned hex code 21 in the ASCII character set, is also hex 21 in the UTF-8 character set. Symbols and characters from another supported character set have a paired set of hexadecimal codes assigned to them in the UTF-8 character set. For example, the letter a with an acute accent (á) is assigned UTF-8 code C3 A1.

The syntax you use to represent these UTF-8 codes in the App Firewall configuration is "xNN" for ASCII characters; "xNNxNN" for non-ASCII characters used in English, Russian, and Turkish; and "xNNxNNxNN" for characters used in Chinese, Japanese, and Korean. For example, if you want to represent a ! in an App Firewall regular expression as a UTF-8 character, you would type x21. If you want to include an á, you would type xC3xA1.

> **Note**:
>
> Normally you do not need to represent ASCII characters in UTF-8 format, but when those characters might confuse a web browser or an underlying operating system, you can use the character's UTF-8 representation to avoid this confusion. For example, if a URL contains a space, you might want to encode the space as \x20 to avoid confusing certain browsers and web server software.

Below are examples of URLs, form field names, and safe object expressions that contain non-ASCII characters that must be entered as PCRE-format regular expressions to be included in the App Firewall configuration. Each example shows the actual URL, field name, or expression string first, followed by a PCRE-format regular expression for it.

- A URL containing extended ASCII characters. Actual URL: `http://www.josénuñez.com Encoded URL: ^http://www[.]jos\xC3\xA9nu\xC3\xB1ez[.]com$`
- Another URL containing extended ASCII characters. Actual URL:`http://www.example.de/trömso.html Encoded URL: ^http://www[.]example\[.]de/tr\\xC3\\xB6mso[.]html$`

A form field name containing extended ASCII characters. Actual Name: `nome_do_usuário Encoded Name: ^nome_do_usu\xC3\xA1rio$`

A safe object expression containing extended ASCII characters. Unencoded Expression `[A-Z]{ 3,6 } ¥\[1-9\][0-9]{ 6,6 }` Encoded Expression: `[A-Z]{ 3,6 } \xC2\xA5[1-9][0-9]{ 6,6 }`

You can find a number of tables that include the entire Unicode character set and matching UTF-8 encodings on the Internet. A useful web site that contains this information is located at the following URL:

```
http://www.utf8-chartable.de/unicode-utf8-table.pl
<!--NeedCopy-->
```

For the characters in the table on this web site to display correctly, you must have an appropriate Unicode font installed on your computer. If you do not, the visual display of the character may be in error. Even if you do not have an appropriate font installed to display a character, however, the description and the UTF-8 and UTF-16 codes on this set of web pages will be correct.

Whitehat WASC signature types for WAF use

September 21, 2020

The NetScaler App Firewall accepts and generates blocking rules for all vulnerability types that the Whitehat scanners generate. However, certain vulnerabilities are most applicable to a web App Firewall. Following are lists of those vulnerabilities, categorized by whether they are addressed by WASC 1.0, WASC 2.0, or best practices signature types.

WASC 1.0 signature types

- HTTP Request Smuggling
- HTTP Response Splitting
- HTTP Response Smuggling
- Null Byte Injection
- Remote File Inclusion
- URL Redirector Abuse

WASC 2.0 signature types

- Abuse of Functionality
- Brute Force
- Content Spoofing
- Denial of Service
- Directory Indexing
- Information Leakage
- Insufficient Anti-automation
- Insufficient Authentication
- Insufficient Authorizatio
- Insufficient Session Expiration
- LDAP Injection
- Session Fixation

Best Practices

- Autocomplete Attribute
- Insufficient Cookie Access Control
- Insufficient Password Strength
- Invalid HTTP Method Usage
- Non-HttpOnly Session Cookie

- Persistent Session Cookie
- Personally Identifiable Information
- Secured Cachable HTTP Messages
- Unsecured Session Cookie

Streaming support for request processing

September 21, 2020

> **Note**:
>
> This feature is available in NetScaler release 10.5.e.

The NetScaler App Firewall now uses request side streaming, which results in a significant performance boost. Instead of buffering the entire request before processing it, the App Firewall now looks at the incoming data, field by field, to inspect the input of each field for any configured security check violation (SQL, XSS, Field Consistency, Field Formats, etc.). As soon as the processing of the data for a field is completed, it is forwarded to the backend while the evaluation continues for the remaining fields. This significantly improves the processing time specially when handling large posts where the forms have large number of fields.

Although the streaming process is transparent to the users, minor configuration adjustments are required due to the following changes:

RegEx Pattern Match: RegEx pattern match is now restricted to 4K for contiguous character string match.

Field Name Match: App Firewall learning engine can only distinguish the first 128 bytes of the name for learning. If a form has multiple fields with names that have identical string match for the first 128 bytes, the learning engine may not be able to distinguish between them. Similarly, the deployed relaxation rule might inadvertently relax all such fields.

Removing white spaces, percent decoding, unicode decoding, and charset conversion which is done during canonicalization is carried out prior to security check inspection. The 128 byte limit is applicable to the canonicalized representation of the field name in UTF-8 character format. The ASCII characters are 1 byte but the UTF-8 representation of the characters in some international languages may range from 1-4 bytes. If each character in the name takes 4 bytes when converted to UTF-8 format, only first 32 characters in the name may be distinguished by the learned rule for such a language.

Field Consistency Check: When the field Consistency check is enabled, all the forms in the session are now stored based on the “as_fid” tag inserted by the App Firewall without consideration for the “action_url.”

- **Mandatory Form tagging for Form Field consistency**: When the field consistency check is enabled, the form tag must be enabled also. The Field Consistency protection might not work if form tagging is turned off.
- **Sessionless Form Field Consistency**: The App Firewall no longer carries out the "GET" to "POST" conversion of forms when sessionless field consistency parameter is enabled. The form tag is required for sessionless field consistency also.
- **Tampering of as_fid**: If a form is submitted after tampering as_fid, it now triggers field consistency violation even if no other field was tampered. In non-streaming requests, this was allowed because the forms could be validated using the "action_url" stored in the session.

Signatures: The signatures now have the following specifications:

- **Location**: It is now a mandatory requirement that location must be specified for each pattern. All patterns in the rule **MUST** have a `<Location>` tag.

- **Fast Match**: All signature rules must have a fast match pattern. If there is no fast match pattern, an attempt will be made to select one if possible. Fast match must be a literal string but some PCRE's can be used for fast match if they contain a usable literal string.

- **Deprecated Locations**: Following locations are no longer supported in signature rules.

  - HTTP_ANY
  - HTTP_RAW_COOKIE
  - HTTP_RAW_HEADER
  - HTTP_RAW_RESP_HEADER
  - HTTP_RAW_SET_COOKIE

XSS/SQL Transform: Raw data is used for transformation because the SQL special characters (single quote('), backslash (), and semicolon (;)), and XSS tags ((<) and(>)) are same in all languages and do not need canonicalization of data. All representations of these characters, such as HTML entity encoding, percent encoding, or ASCII are evaluated for transform operation.

The App Firewall no longer inspects both the attribute name and value for the XSS transform operation. Now only XSS attribute names are transformed when streaming is engaged.

Processing XSS Tags: As part of the streaming changes in 10.5.e build onwards, the App Firewall processing of the Cross-site Scripting tags has changed. In earlier releases, presence of either open bracket (<), or close bracket (>), or both open and close brackets (<>) was flagged as Cross-site Scripting Violation. The behavior has changed in 10.5.e build onwards. Presence of only the open bracket character (<), or only the close bracket character (>) is no longer considered as an attack. It is when an open bracket character (<) is followed by a close bracket character (>), the Cross-site scripting attack gets flagged. Both characters must be present in the right order (< followed by >) to trigger Cross-site scripting violation.

> **Note**
>
> **Change in SQL violation log Message**: As part of the streaming changes in 10.5.e build onwards, we now process the input data in blocks. RegEx pattern matching is now restricted to 4K for contiguous character string matching. With this change, the SQL violation log messages might include different information compared to the earlier builds. The keyword and special character in the input could be separated by a large number of bytes. We now keep track of the SQL keywords and special strings when processing the data, instead of buffering the entire input value. In addition to the field name, the log message now includes the SQL keyword, or the SQL special character, or both the SQL keyword and the SQL special character, as determined by the configured setting. The rest of the input is no longer included in the log message, as shown in the following example:
>
> **Example:**
>
> In 10.5, when the App Firewall detects the SQL violation, the entire input string might be included in the log message, as shown below:
>
> SQL Keyword check failed for field **text=”select a name from testbed1\;\(\;\)”.*<blocked>**
>
> In 11.0, we log only the field name, keyword and special character (if applicable) in the log message, as shown below:
>
> SQL Keyword check failed for field `text="select(;)"<blocked>`
>
> This change is applicable to requests that contain **application/x-www-form-urlencoded**, or **multipart/form-data**, or **text/x-gwt-rpc** content-types. Log messages generated during processing of **JSON** or **XML** payloads are not affected by this change.

RAW POST Body: The security check inspections are always done on RAW POST body.

Form ID: The App Firewall inserted “as_fid” tag, which is a computed hash of the form, will no longer be unique for the user session. It will now have an identical value for a specific form irrespective of the user or the session.

Charset: If a request does not have a charset, the default charset specified in the application profile is used when processing the request.

Counters:

Counters with prefix “se_” and “appfwreq_” are added to track the streaming engine and the App Firewall streaming engine request counters respectively.

nsconsmg -d statswt0 -g se_err_

nsconsmg -d statswt0 -g se_tot_

nsconsmg -d statswt0 -g se_cur_

nsconsmg -d statswt0 -g appfwreq_err_

nsconsmg -d statswt0 -g appfwreq_tot_

nsconsmg -d statswt0 -g appfwreq_cur_

_err counters: indicate the rare event which should have succeeded but failed due to either memory allocation problem or some other resource crunch.

_tot counters: ever increasing counters.

_cur counters: counters indicating current values that keep changing based on usage from current transactions.

Tips:

- The App Firewall security checks should work exactly the same as before.
  > - There is no set ordering for the processing of the security checks.
- The response side processing is not affected and remains unchanged.
- Streaming is not engaged if CVPN is used.

> **Important**
>
> **Calculating the Cookie length:** In release 10.5.e (in a few interim enhancement builds prior to 59.13xx.e build) as well as in the 11.0 release (in builds prior to 65.x), App Firewall processing of the Cookie header was changed. In those releases, every cookie is evaluated individually, and if the length of any one cookie received in the Cookie header exceeds the configured BufferOverflowMaxCookieLength, the Buffer Overflow violation is triggered. As a result of this change, requests that were blocked in 10.5 and earlier release builds might be allowed, because the length of the entire cookie header is not calculated for determining the cookie length. In some situations, the total cookie size forwarded to the server might be larger than the accepted value, and the server might respond with "400 Bad Request".
>
> **Note that this change has been reverted. The behavior in the 10.5.e ->59.13xx.e and subsequent 10.5.e enhancement builds as well as in the 11.0 release 65.x and subsequent builds is now similar to that of the non-enhancement builds of release 10.5. The entire raw Cookie header is now considered when calculating the length of the cookie. Surrounding spaces and the semicolon (;) characters separating the name-value pairs are also included in determining the cookie length.**

Trace HTML requests with security logs

September 21, 2020

> **Note**:
>
> This feature is available in NetScaler release 10.5.e.

Troubleshooting a problem, which requires analysis of data received in the client request can be quite challenging specially when there is heavy traffic flowing through the box. Diagnosing issues which may affect the functionality or security of the application require a quick response.

The NetScaler now offers the option to isolate traffic for a specific App Firewall profile and collect nstrace for the HTML requests that trigger a log or block action or malformed requests that might be causing reset or aborts. The nstrace collected in –appfw mode will include details of the entire request including the App Firewall generated log messages. You can use "Follow TCP stream" in the trace to view the details of the individual transaction including headers, payload, as well as the corresponding log message, together in the same screen.

This gives you a comprehensive overview regarding your traffic. Having a detailed view of the request, payload, and associated log records can be very useful to analyze security check violation. You can easily identify the pattern that is triggering the violation. If the pattern should be allowed, you can take a decision to modify the configuration and/or add a relaxation rule.

This enhancement helps in troubleshooting the NetScaler and offers the following benefits:

1. **Isolate traffic for specific profile**: This enhancement can be quite useful when you need to isolate traffic for only one profile or specific transactions of a profile for troubleshooting. You no longer have to skim through the entire data collected in the trace or need special filters to isolate requests that are of interest to you which can be tedious especially with heavy traffic. You now have the option to view only the data that you are interested in.
2. **Collect data for specific requests**: The trace can be collected for a specified duration. You can collect trace for only a couple of requests to isolate, analyze, and debug specific transactions if needed.
3. **Identify resets or aborts**: Unexpected closing of connections are not easily visible. The trace collected in –appfw mode captures a reset or an abort, triggered by the App Firewall. This allows a quicker isolation of issue when you do not see a security check violation message. Malformed requests or other non-RFC compliant requests terminated by App Firewall will now be easier to identify.
4. **View decrypted SSL traffic**: HTTPS traffic is captured in plain text to allow for easier troubleshooting.
5. **Provides comprehensive view**: Allows you to look at the entire request at the packet level, check the payload, look at the logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n etc), they are easy to discover in the trace.
6. **Modify configuration**: The debugging can provide useful information to decide if the observed

behavior is the correct behavior or the configuration should be modified.

7. **Expedite response time**: Faster debugging on target traffic can improve the response time to provide explanations and/or root cause analysis by Citrix engineering and support team.

Please see any task topic in eDocs for documenting tasks. Manual Configuration by using the command line interface

To configure debug tracing for a profile by using the command line interface

Step 1. Enable tracing for the profile. You can use the show command to verify the configured setting.

- `set appfw profile <profile> -trace ON`

Step 2. Start collecting trace. You can continue to use all the options which are applicable for the nstrace command.

- `start nstrace -mode APPFW`

Stop collecting the trace

- `stop nstrace`

Location of the trace: The nstrace is stored in a time-stamped folder which is created in the /var/nstrace directory and can be viewed using wireshark. You can tail the /var/log/ns.log to see the log messages providing details regarding the location of the new trace.

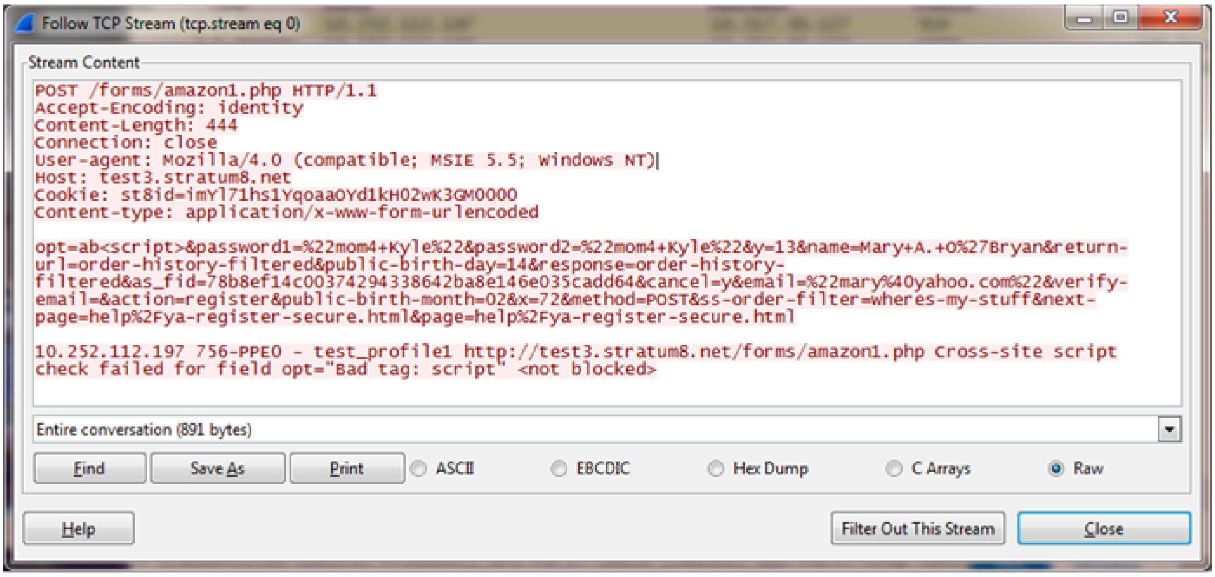
Tips:

- When –appfw mode option is used, the nstrace will only collect the data for the profile(s) for which "trace" was enabled.
- Enabling trace on the profile will not automatically start collecting the traces till you explicitly execute the "start nstrace" command to collect the trace.
- Although, enabling trace on a profile may not have any adverse effect on the performance of the App Firewall but you may want to enable this feature only for the duration for which you want to collect the data. It is recommended that you turn the –trace flag off after you have collected the trace. This will prevent the risk of inadvertently getting data from profiles for which you had enabled this flag in the past.
- The Block or Log action must be enabled for the security check for the transaction record to be included in the nstrace.
- Resets and aborts will be logged independently of security checks actions when trace is "On" for the profile(s).
- This feature is only applicable for troubleshooting the requests received from the client. The traces in –appfw mode do not include the responses received from the server.
- You can continue to use all the options which are applicable for the nstrace command. For example,

```
start nstrace -tcpdump enabled -size 0 -mode appFW
```

- If a request triggers multiple violations, the nstrace for that record will include all the corresponding log messages.
- CEF log message format is supported for this functionality.
- Signature violations triggering block and/or log action for request side checks will also be included in the trace.
- Only HTML (non-XML) requests are collected in the trace.

Example of a Log record in the trace:



```
POST /forms/amazon1.php HTTP/1.1
Accept-Encoding: identity
Content-Length: 444
Connection: close
User-agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT)
Host: test3.stratum8.net
Cookie: st8id=imYl71hs1YqoaaOYd1kH02wK3GM0000
Content-type: application/x-www-form-urlencoded

opt=ab<script>&password1=%22mom4+Kyle%22&password2=%22mom4+Kyle%22&y=13&name=Mary+A.+O%27Bryan&return-
url=order-history-filtered&public-birth-day=14&response=order-history-
filtered&as_fid=78b8ef14c00374294338642ba8e146e035cadd64&cancel=y&email=%22mary%40yahoo.com%22&verify-
email=&action=register&public-birth-month=02&x=72&method=POST&ss-order-filter=wheres-my-stuff&next-
page=help%2Fya-register-secure.html&page=help%2Fya-register-secure.html

10.252.112.197 756-PPE0 - test_profile1 http://test3.stratum8.net/forms/amazon1.php Cross-site script
check failed for field opt="Bad tag: script" <not blocked>
```

App Firewall support for cluster configurations

September 21, 2020

> **Note**:
>
> App Firewall support for Striped and partially striped configurations was introduced in NetScaler release 11.0.

A cluster is a group of NetScaler appliances that are configured and managed as a single system. Each appliance in the cluster is called a node. Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations. The App Firewall is fully supported in all configurations.

The two main advantages of striped and partially striped virtual server support in cluster configurations are the following:

1. Session failover support—Striped and partially striped virtual server configurations support session failover. The advanced App Firewall security features, such as Start URL Closure and the Form Field Consistency check, maintain and use sessions during transaction processing. In ordinary high availability configurations, or in spotted cluster configurations, when the node that is processing the App Firewall traffic fails, all the session information is lost and the user has to reestablish the session. In striped virtual server configurations, user sessions are replicated across multiple nodes. If a node goes down, a node running the replica becomes the owner. Session information is maintained without any visible impact to the user.
2. Scalability—Any node in the cluster can process the traffic. Multiple nodes of the cluster can process the incoming requests served by the striped virtual server. This improves the App Firewall's ability to handle multiple simultaneous requests, thereby improving the overall performance.

Security checks and signature protections can be deployed without the need for any additional cluster-specific App Firewall configuration. You just do the usual App Firewall configuration on the configuration coordinator (CCO) node for propagation to all the nodes.

> **Note**:
>
> The session information is replicated across multiple nodes, but not across all the nodes in the striped configuration. Therefore, failover support accommodates a limited number of simultaneous failures. If multiple nodes fail simultaneously, the App Firewall might lose the session information if a failure occurs before the session is replicated on another node.

Highlights

- App Firewall offers scalability, high throughput, and session failover support in cluster deployments.
- All App Firewall security checks and signature protections are supported in all cluster configurations.
- Character-Maps are not yet supported for a cluster. The learning engine recommends Field-Types in learned rules for the Field Format security check.
- Stats and learned rules are aggregated from all the nodes in a cluster.
- Distributed Hash Table (DHT) provides the caching of the session and offers the ability to replicate session information across multiple nodes. When a request comes to the virtual server, the NetScaler appliance creates App Firewall sessions in the DHT, and can also retrieve the session information from the DHT.
- Clustering is licensed with the Enterprise and Platinum licenses. This feature is not available with the Standard license.

Debugging and troubleshooting

September 21, 2020

Refer to the following troubleshooting and debugging information related to each of the App Firewall functionality:

- Application Firewall - High CPU
- Memory
- Large File Upload Failure
- Learning
- Signatures
- Trace log
- Miscellaneous

High CPU

September 21, 2020

Following are some of the functionality and high CPU related debugging issues encoutered and the best practices to follow when working with App Firewall:

• Check Policy hits, Bindings, Network configuration, App Firewall configuration

- Identify misconfiguration
- Identify *vserver* that is serving the affected traffic

• Inspect logs in the following log files for security violations and recent configuration changes

- `/var/log/ns.log`
- `/var/nslog/import.log`
- `/var/nslog/aslearn.log`

tail -f /var/log/ns.log	grep APPFW_SIGNATURE_MATCH

-

Example:

```
Jun 13 01:11:09 <local0.info> 10.217.31.98 CEF:0|Citrix|NetScaler|NS11
    .0|APPFW| APPFW_SIGNATURE_MATCH|6|src=10.217.253.62 spt=61141 method
```

```
=GET request= http://aaron.stratum8.net/FFC/wwwboard/passwd.txt msg=
Signature violation rule ID 807: web-cgi /wwwboard/passwd.txt access
 cn1=140 cn2=841 cs1=pr_ffc cs2=PPE0 cs3=
OyTgjbXBqcpBFeENKDlde3OkMQ00001 cs4=ALERT cs5=2015 cs6=web-cgi act=
not blocked
<!--NeedCopy-->
```

• **Isolate the traffic that is effected**

- Isolate the profile
- Isolate the security check
- Isolate the URL, vserver and traffic parameters

• **Conditional profile level trace helps identify the traffic and violation records**

- `set appfw profile <profile> -trace ON`
- `start nstrace -mode APPFW -size 0`
- `stop nstrace`

Note: Ensure that the trace is collected with -size 0 option.

• **Check appfw, dht, IP reputation activity counters**

- `nsconmsg -g as_ -g appfwreq_ -g iprep -d current`

• **Monitor window size for resets in connection**

- Appfw sets the window size to 9845 when NetScaler resets the connection due to an invalid http message.

Examples:

- Malformed request received - connection reset
- High CPU related issues
- Check data sheets for system limits
- Inspect for cpu usage, appfw, DHT and memory related activity. Monitor appfw sessions
- nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -g mem_AS_OBJ -g mem_AS_COMPONENT -d current

• Monitor memory allocated and freed from App Firewall components and objects during the target time period. It helps in isolating the protection leading to high CPU usage.

- Profiler output

- Observe logs

- **Isolate appfw check leading to high CPU**

- startURLClosure

- Formfiledconsistency

- CSRF

- Cookie protections

- Referer header check

Ascertain that autoupdate of signatures is not leading to high CPU (Disable to confirm).

Memory

September 21, 2020

Following are some of the best practices to follow when encountered with App Firewall usage memory related issues:

nsconmsg command usage:

• Look for global memory statistics to ascertain that there is enough memory in the system and there are no memory allocation failures by executing the following command:

```
* *- nsconmsg -d memstats
```

• Observe current allocated and maximum memory limits for appsecure, IP reputation, cache and compression by executing the following command:

- `nsconmsg -d memstats | egrep -i APPSECURE|IPREP|CACHE|CMP`

• Check appfw, DHT, IP reputation activity counters by executing the following command:

- `nsconmsg -g as -g appfwreq_ -g iprep -d current`

• Check all App Firewall error counters by executing the following command:

- `nsconmsg -g as_ -g appfwreq_ -g iprep_ -d stats | grep err`

• Check all system error counters by executing the following command:

- `nsconmsg -g err -d current`

• Inspect for CPU, APPFWREQ, AS and DHT counters by executing the following command:

  - `nsconmsg -g cc_cpu_use -g appfwreq -g as -g dht -d current`

• Check the configured Cache memory by executing the following command:

  - `show cacheparameter`

• Check the configured memory by executing the following command:

  - `nsconmsg -d memstats | egrep -i CACHE`

• Identify distribution of memory in App Firewall components and objects:

Display AS_OBJ_ memory:

- `nsconmsg -K newnslog -d stats | grep AS_OBJ | egrep -v AppFW_cpu0|total | sort -k3`

Display AS_COMPONENT_ memory:

- `nsconmsg -K newnslog -d stats | grep AS_COMPONENT | egrep -v AppFW_cpu0 |total | sort -k3`

• Check for number of alive sessions by executing the following command:

Monitor/plot active session counts:

- `nsconmsg -g as_alive_sessions -d current`

Monitor/plot total allocated, free, updated sessions:

- `nsconmsg -g as_tot_alloc_sessions -g as_tot_free_sessions -d current`
- `nsconmsg -g as_tot_update_sessions -d current`

If required, reduce session timeout to ensure that session limits are not used by executing the following command:

- `set appfwsettings -sessionTimeout <300>`

If required, set maximum lifetime of session by executing the following command:

- `set appfwsettings -sessionLifetime <7200>`

Checking allocated and used memory

To check the total allocated memory and used memory:

- Use the **nsconmsg –d memstats** command. Observe the **MEM_APPSECURE** field.
- Use the **stat appfw** command to obtain meory consumption information.

App Firewall does not automatically delete the logs after certain period of time or size.

- `All AppFw logs are archived in the */var/log/ns.log*` file. The *ns.log* file performs the rollover task.

For more information, refer to the following link: <http://support.citrix.com/article/CTX121898>

Increasing App Firewall memory:

- There is no CLI option to increase App Firewall memory. App Firewall memory is platform-specific.

- You may use the *nsapimgr* option to increase memory but it is not recommended.

The max allowed memory for App Firewall is determined by the platform and disabling IC does not impact memory allocation.

Large file upload failure

September 21, 2020

When you encounter large file upload failures, ensure that you check the following:

– Misconfigured application firewall postbody limit
– Enabled file upload scanning leading to increased processing time.
– Hitting system limits.

For payloads greater than 20 MB, Citrix recommends you to enable streaming on the application firewall profile. Also, you must ensure the back-end server supports chunked requests before enabling streaming.

Since release 11.0, the streaming flag can be enabled on per profile basis to avoid buffering by executing the following command:

```
set appfw profile <profile name> -streaming on
```

Learning

September 21, 2020

Following are some of the best practices recommended when encountered with Learning functionality issues:

Aslearn process:

• Verify that the process *aslearn* is running.

• Check top command output

• Check output of ps command by executing the following command:

- `ps -ax | grep aslearn | grep -v "grep”`

Example:

```
root@ns\# ps -ax | grep aslearn | grep -v "grep"

```

```
1439  ??  Ss      0:03.86 /netscaler/aslearn -start -f /netscaler/aslearn.conf
<!--NeedCopy-->
```

- Identify recent configuration commands executed prior to the observed problem by verifying the *ns.log* file:
    - /var/log/ns.log
- Inspect aslearn logs to check for aslearn messages:
    - /var/log/aslearn.log
- Isolate the profile and security check that is effected
- Identify the GUI and CLI command which is failing by executing the following command:
    - `show appfw learningdata <profileName> <securityCheck>`

Examples:

- show learningdata test_profile starturl
- show learningdata test_profile crosssiteScripting
- show learningdata test_profile sqLInjection
- show learningdata test_profile csRFtag
- show learningdata test_profile fieldformat
- show learningdata test_profile fieldconsistency

- Perform integrity check of sqlite from bsd shell prompt:
    - `nsshell ## sqlite3 /var/nslog/asl/<profile_name_in_lowercase>.db 'pragma integrity_check;'`

Examples:

```
- root@ns# sqlite3 /var/nslog/asl/tsk0247284.db 'pragma integrity_check;'
Ok
<!--NeedCopy-->
```

- Deploy or remove rules to start learning again:
    - If 2000 learn items (per protection) are reached, you cannot start learning any more for that protection
    - If 20 MB size is reached for the database, stop learning for all protections
    - Restart aslearn process

`*/netscaler/aslearn -start -f/netscaler/aslearn.conf*`

• Check the space in the /var folder by executing the following:

- du -h /var

• Check the learning threshold limits by executing the following command:

- `show appfwlearningsettings <profile_name> <securityCheck>`

• Collect learned data by executing the following command:

- `export appfwlearningdata <profile_name> <securityCheck>`

• Ascertain that learned data is uploaded in the collector.

Signatures

September 21, 2020

Getting started with signatures

To add signature:

1. Select the **Default**-signature and click **add** to make a copy.
2. Give a meaningful name. The new sig object is added as a User-Defined object.
3. Enable the target rules that are pertinent to your specific need.

– The rules are disabled by default.
– more rules require more processing

4; Configure the actions:

– Block and Log actions are enabled by default. Stats is another option

5; Set the signature to be used by your profile.

Tips for using signatures

• Optimize the processing overhead by enabling only those signatures that are applicable for protecting your application.

• Every pattern in the rule must match to trigger a signature match.

• You can add your own customized rules to inspect incoming requests to detect various types of attacks, such as SQL injection or cross-site scripting attacks.

- You can also add rules to inspect the responses to detect and block leakage of sensitive information such as credit card numbers.
- Add multiple security check conditions to create your own customized check.

Best practices for using signatures

Following are some of the best practices you can follow when encountered with issues related to Signatures:

- Verify that the import command has succeeded on primary as well as secondary.
- Verify that CLI and GUI outputs are consistent.
- Check ns.log to identify any errors during signature import and auto update.
- Check if the DNS name server is configured properly.
- Check schema version incompatibility.
- Check if the device is unable to access the Signature Update URL hosted on AWS for auto-update.
- Check for the version mismatch between Default signature and user-added ones.
- Check for version mismatch between signature objects on the primary and secondary nodes.
- Monitor for High CPU Utilization (disable auto-update to rule out issue with signature update).

Trace Log

September 21, 2020

To record trace logs:

1. Enable tracing for the profile. You can use the show command to verify the configured setting.

```
set appfw profile <profile> -trace ON
```

2. Start collecting trace. You can continue to use all the options which are applicable for the nstrace command.

```
start nstrace -mode APPFW
```

3. Stop collecting the trace

```
stop nstrace
```

Location of the trace: The nstrace is stored in a time-stamped folder which is created in the */var/nstrace* directory and can be viewed using wireshark. You can tail the */var/log/ns.log* file to see the log messages providing details regarding the location of the new trace.

Advantages of trace logs:

– Isolate traffic for specific profile

– Collect data for specific requests

– Identify resets or aborts

- View decrypted SSL traffic: HTTPS traffic is captured in plain text to allow for easier troubleshooting.

– Provides comprehensive view: Allows you to look at the entire request at the packet level, check the payload, view logs to check what security check violation is being triggered and identify the match pattern in the payload. If the payload consists of any unexpected data, junk strings, or non-printable characters (null character, \r or \n etc), they are easy to discover in the trace.

– Expedite response time: Faster debugging on target traffic to do root cause analysis.

Miscellaneous

September 21, 2020

Following are the resolutions for some of the issues that you might encounter when using App Firewall.

• App Firewall sets window size to 9845 when resetting connection for invalid http messages.

– Malformed request received - connection reset [Client/Server sending invalid content-length header]

– Unknown content-type in request headers

• System Limit: the application appears frozen

– Occurs when maximum session limit is reached. (100K)

– Less system memory for operation.

• IP Reputation feature not working

– The iprep process takes about five minutes to start after you enable the reputation feature. The IP reputation feature might not work for that duration.

• Unexpected App Firewall violations being triggered

– Session timeout has a default value of 900 seconds. If session timeout is set to a low value, browser may trigger false positives for checks which rely on sessionization (e.g CSRF, FFC). Check for session timeout and look at the session ID (cs3 in CEF logs). If the sessionID is different, the session timeout could be the reason.

– If form is dynamically generated by javascript, it may trigger false FFC violations.

• Empty field name in FFC violation logs (prior to 11.0 release)

This may be seen in scenarios where we come across a form field which is not in the forms in our session.

Scenarios where this may occur:

– The session has timed out from when the form was sent to the client and when it was received.

– The form was generated on the client side using a java script.

References

September 21, 2020

Refer to the following additional reosources for more information and useful tips when using App Firewall:

- HowNetScaler App Firewall Modifies Application DataTraffic

– Conditional headers modified by App Firewall.

– Integrated caching and App Firewall interoperability.

- TraceHTML Requests with App Firewall Security Violation Logs on NetScalerAppliance

–Isolating request and debugging the end-to-end transaction.

- Top Level Protection
    - security relaxations

–Information about configuring and deploying.

    - Application
    - Firewall
    - Logs

– Details regarding anatomy of the App Firewall log messages.

- `<https://regex101.com/>`

– Configuring Regular expressions.

- Datasheets

– Using recommended memory and CPU for system.

– Ensuring enough memory for App Firewall and configuring cache limit appropriately.

Cache Redirection

September 21, 2020

In a typical deployment, different clients ask web servers for the same content repeatedly. To relieve the origin web server of processing each request, a NetScaler appliance with cache redirection enabled can serve this content from a cache server instead of from the origin server.

The NetScaler appliance analyzes incoming requests, sends requests for cacheable data to cache servers, and sends non-cacheable requests and dynamic HTTP requests to origin servers.

Cache redirection is a policy-based feature. By default, requests that match a policy are sent to the origin server, and all other requests are sent to a cache server. For testing or maintenance, you might want to skip policy evaluation and direct all requests to the cache or to the origin server.

You can combine content switching with cache redirection to cache selective content and serve content from specific cache servers for specific types of requested content.

A NetScaler appliance configured for cache redirection can be deployed at the edge of a network, in front of the origin server, or anywhere along the network backbone. In an edge deployment, commonly used by Internet Service Providers (ISPs), cable companies, content delivery distribution networks, and enterprise networks, the NetScaler appliance resides directly in front of the clients. In a server-side deployment, the NetScaler appliance is closer to the origin servers.

Cache redirection is used most commonly with the HTTP service type, but it also supports the secure HTTPS protocol.

Cache redirection policies

September 21, 2020

A cache redirection virtual server applies cache redirection policies to each incoming request. By default, if a request matches one of the configured policies, it is considered non-cacheable, and the NetScaler appliance sends it to the origin server. Other requests are sent to a cache server. This behavior can be reversed, so that requests that match configured cache redirection policies are sent to cache servers.

The appliance provides a set of policies for cache redirection. If these built-in policies are not adequate for your deployment, you can configure user-defined cache redirection policies.

> **Note**
>
> Once you have determined which built-in cache redirection policies to use, or have created user-

> defined policies, proceed with configuring cache redirection. To use this feature, you must configure at least one cache redirection virtual server, and, for normal operation, you must bind at least one cache redirection policy to that virtual server.

Built-in cache redirection policies

September 21, 2020

The NetScaler appliance provides built-in cache redirection policies that handle typical cache requests. These policies are based on HTTP methods, the URL or URL tokens of the incoming request, the HTTP version, or the HTTP headers and their values in the request.

Built-in cache redirection policies can be directly bound to a virtual server and do not need further configuration.

Cache redirection policies use two types of appliance expressions languages, classic and default syntax. For more information about these languages, see Policies and expressions.

Built-in classic cache redirection policies

Built-in cache redirection policies based on classic expressions are called *classic cache redirection policies*. For a complete description of classic expressions and how to configure them, see Policies and Expressions.

The classic cache redirection policies evaluate basic characteristics of traffic and other data. For example, classic cache redirection policies can determine whether an HTTP request or response contains a particular type of header or URL.

The NetScaler appliance provides the following built-in classic cache redirection policies:

Built-In Policy Name	Description
bypass-non-get	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.

Built-In Policy Name	Description
bypass-dynamic-url	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: cgi, asp, exe, cfm, ex, shtml, or htx. Also bypass the cache if the URL starts with any of the following: /cgi-bin/, /bin/, or /exec/.
bypass-urltokens	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie	Bypass the cache for any URL that has a cookie header and an extension other than .png or .jpg.

Built-in default syntax cache redirection policies

Built-in cache redirection policies based on default syntax expressions are called *default syntax cache redirection policies*. For a complete description of default syntax expressions and how to configure them, see Policies and Expressions.

In addition to the same types of evaluations done by classic cache redirection policies, default syntax cache redirection policies enable you to analyze more data (for example, the body of an HTTP request) and to configure more operations in the policy rule (for example, directing the request to either cache or origin server).

NetScaler appliances provide the following two built-in actions for the default syntax cache redirection policies:

- CACHE
- ORIGIN

As implied by their names, they direct the request to the cache server or the origin server, respectively.

> **Note**
>
> If you are using the built-in default syntax cache redirection policy, you cannot modify the action.

The NetScaler appliance provides the following built-in default syntax cache redirection policies:

Built-In Policy Name	Description
bypass-non-get_adv	Bypass the cache if the request uses an HTTP method other than GET.
bypass-cache-control_adv	Bypass the cache if the request header contains a Cache-Control: no-cache or Cache-Control: no-store header, or if the HTTP request contains a pragma header.
bypass-dynamic-url_adv	Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions: cgi, asp, exe, cfm, ex, shtml, or htx. Also bypass the cache if the URL starts with any of the following: /cgi-bin/, /bin/, or /exec/.
bypass-urltokens_adv	Bypass the cache because the request is dynamic, as indicated by one of the following tokens in the URL: ?, !, or =.
bypass-cookie_adv	Bypass the cache for any URL that has a cookie header and an extension other than .png or .jpg.

Display the built-in cache redirection policies

You can display the available cache redirection polices by using the command line interface or the configuration utility.

Display the built-in cache redirection policies by using the CLI

At the command prompt, type:

`show cr policy [<policyName>]`

Example:

```
> show cr policy
1)        Cache-By-Pass RULE: NS_NON_GET            Policy:bypass-non-get
2)        Cache-By-Pass RULE: (NS_CACHECONTROL_NOSTORE || NS_CACHECONTROL_NOCACHE || NS_HEADER_PRAGMA)     Policy:bypass-cache-control
3)        Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE || NS_EXT_CFM || NS_EXT_EX || NS_EXT_SHTML || NS_EXT_HTX) || (
   NS_URL_PATH_CGIBIN || NS_URL_PATH_EXEC || NS_URL_PATH_BIN)
   Policy:bypass-dynamic-url
4)       Cache-By-Pass RULE: NS_URL_TOKENS       Policy:bypass-
   urltokens
5)       Cache-By-Pass RULE: (NS_HEADER_COOKIE && NS_EXT_NOT_GIF &&
   NS_EXT_NOT_JPEG)     Policy:bypass-cookie
 Done
<!--NeedCopy-->
```

Display the built-in cache redirection policies by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies. The configured cache redirection policies appear in the details pane.
2. Select one of the configured policies to view details.

Configure a cache redirection policy

September 21, 2020

A cache redirection policy includes one or more expressions (also called *rules*). Each expression represents a condition that is evaluated when the client request is compared to the policy.

You do not explicitly configure actions for cache redirection policies. By default, the NetScaler appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the classic policy format are called *classic cache redirection policies*. Each such policy has a name and includes a classic expression or a set of classic expressions that are combined by using logical operators.

For classic cache redirection policies, you do not explicitly configure actions for the policies. By default, the NetScaler appliance considers any request that matches a policy to be non-cacheable and directs the request to the origin server instead of the cache.

Cache redirection policies based on the newer policy format are called *Advanced redirection policies*. Such policy has a name and includes a default syntax expression, or a set of default syntax expressions that are combined by using logical operators, and the following built-in actions:

- CACHE
- ORIGIN

For more information about classic expressions and default syntax expressions, see Policies and Expressions.

Add a cache redirection policy by using the CLI

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

```
-  add cr policy <policyName> **-rule** <expression>
-  show cr policy [<policyName>]
<!--NeedCopy-->
```

Examples:

Policy with a simple expression:

```
> add cr policy Policy-CRD-1 -rule "REQ.HTTP.URL != /*.jpeg"
 Done
> show cr policy Policy-CRD-1
        Cache-By-Pass RULE: REQ.HTTP.URL != '/*.jpeg'   Policy:Policy-CRD-1
 Done
<!--NeedCopy-->
```

Policy with a compound expression:

```
> add cr policy Policy-CRD-2 -rule "REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == /*.cgi || REQ.HTTP.URL != /*.png)"
 Done
> show cr policy Policy-CRD-2
        Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == '/*.cgi' || REQ.HTTP.URL != '/*.png')   Policy:Policy-CRD-2
 Done
<!--NeedCopy-->
```

Policy that evaluates a header:

```
> add cr policy Policy-CRD-3 -rule "REQ.HTTP.HEADER If-Modified-Since EXISTS"
 Done
> show cr policy Policy-CRD-3
        Cache-By-Pass RULE: REQ.HTTP.HEADER If-Modified-Since EXISTS      Policy:Policy-CRD-3
 Done
```

```
<!--NeedCopy-->
```

Add a default syntax cache redirection policy by using the CLI

At the command prompt, type the following commands to add a cache redirection policy and verify the configuration:

```
-  add cr policy <policyName> **-rule** <expression> [-action<string>]
    [-logAction<string>]
-  show cr policy [<policyName>]
<!--NeedCopy-->
```

Examples:

Policy with a simple expression:

```
> add cr policy crpol1 -rule !(HTTP.REQ.URL.ENDSWITH(".jpeg” )) -action
     origin
 Done
> show cr policy crpoll
         Policy: crpol1 Rule: !(HTTP.REQ.URL.ENDSWITH(".jpeg")) Action:
             ORIGIN
 Done
<!--NeedCopy-->
```

Policy with a compound expression:

```
> add cr policy crpol11 -rule "http.req.method.eq(post) && (HTTP.REQ.
    URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))"  -action
    cache
 Done
> show cr policy crpol11
         Policy: crpol11    Rule: http.req.method.eq(post) && (HTTP.REQ
             .URL.ENDSWITH(".png") || HTTP.REQ.URL.ENDSWITH(".cgi"))
             Action: CACHE
 Done
<!--NeedCopy-->
```

Policy that evaluates a header:

```
> add cr policy crpol12 -rule http.req.header("If-Modified-Since").
    exists -action origin
 Done
> show cr policy crpol12
```

```
        Policy: crpol12    Rule: http.req.header("If-Modified-Since").
            exists    Action: ORIGIN
 Done
<!--NeedCopy-->
```

Modify or remove a cache redirection policy by using the CLI

- To modify a cache redirection policy, use the set cr policy command, which is just like add cr policy command, except that you enter the name of an existing policy.
- To remove a policy, use the `rm cr policy` command, which accepts only the <name> argument. If the policy is bound to a virtual server, you have to unbind the policy, before you can remove it.

For the details of unbinding a cache redirection policy, see "Unbind a policy from a cache redirection virtual server."

Configure a cache redirection policy with a simple expression by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies.
2. In the details pane, click Add.
3. In the Create Cache Redirection Policy dialog box, in the Name* text box, type the name of the policy, and then in the Expression area, click Add.
4. To configure a simple expression, enter the expression. Following is an example of an expression that checks for a .jpeg extension in a URL:

    - Expression Type-General
    - Flow Type -REQ
    - Protocol -HTTP
    - Qualifier -URL
    - Operator - !=
    - Value* - /*.jpeg

    The simple expression in the following example checks for an If-Modified-Since header in a request:

    - Expression Type -General
    - Flow Type -REQ
    - Protocol -HTTP
    - Qualifier -HEADER
    - Operator -EXISTS
    - Header Name -If-Modified-Since

5. When you are finished entering the expression, click OK or Create, and then click Close.

Configure a cache redirection policy with a compound expression by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Policies.
2. In the details pane, click Add.
3. In the Name text box, enter a name for the policy.

   The name can begin with a letter, number, or the underscore symbol, and can consist of from one to 127 letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at (@), equals (=), and underscore (_) symbols. You should choose a name that will make it easy for others to tell what type of content this policy was created to detect.

4. Choose the type of compound expression that you want to create. Your choices are:
   - **Match Any Expression**. The policy matches the traffic if one or more individual expressions match the traffic.
   - **Match All Expressions**. The policy matches the traffic only if every individual expression matches the traffic.
   - **Tabular Expressions**. Switches the Expressions list to a tabular format with three columns. In the rightmost column, you place one of the following operators:
     - The AND [&&] operator, to require that, to match the policy, a request must match both the current expression and the following expression.

The OR [	] operator, to require that, to match the policy, a request must match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.

–

You can also group expressions in nested subgroups by selecting an existing expression and clicking one of the following operators:

- The BEGIN SUBGROUP [+ ( ] operator, which tells the NetScaler appliance to begin a nested subgroup with the selected expression. (To remove this operator from the expression, click -(.)

    - The END SUBGROUP [+)] operator, which tells the NetScaler appliance to end the current nested subgroup with the selected expression. (To remove this operator from the expression, click -) .)

  - **Advanced Free-Form**. Switches off the Expressions Editor entirely and turns the Expressions list into a text area in which you can type a compound expression. This is both the most powerful and the most difficult method of creating a policy expression, and is recommended only for those thoroughly familiar with the NetScaler appliance classic expressions language.

    For more information about creating classic expressions in the Advanced Free-Form text area, see "Configuring Classic Policies and Expressions".

    Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that you want to use it.

5. If you chose Match Any Expression, Match All Expressions, or Tabular Expressions, click **Add** to display the Add Expression dialog box.

   You should leave the expression type set to General for cache redirection policies.

6. In the Flow Type drop-down list, choose a flow type for your expression.

   The flow type determines whether the policy examines incoming or outgoing connections. You have two choices:

   - **REQ**. Configures the NetScaler appliance to examine incoming connections, or requests.
   - **RES**. Configures the appliance to examine outgoing connections, or responses.

7. In the Protocol drop-down list, choose a protocol for your expression.

   The protocol determines the type of information that the policy examines in the request or response. Depending upon whether you chose REQ or RES in the previous drop-down list, either all four or only three of the following choices are available:

   - **HTTP**. Configures the appliance to examine the HTTP header.
   - **SSL**. Configures the appliance to examine the SSL client certificate. Available only if you chose REQ (requests) in the previous drop-down list.
   - **TCP**. Configures the appliance to examine the TCP header.
   - **IP**. Configures the appliance to examine the source or destination IP address.

8. Choose a qualifier for your expression from the Qualifier drop-down list.

   The contents of the Qualifier drop-down list depend on which protocol you chose. The following table describes the choices available for each protocol.

   Table 1. Cache Redirection Policy Qualifiers Available for Each Protocol

Protocol	Qualifier	Definition
HTTP	METHOD	HTTP method used in the request.
-	URL	Contents of the URL header.
-	URLTOKENS	URL tokens in the HTTP header.
-	VERSION	HTTP version of the connection.
-	HEADER	Header portion of the HTTP request.
-	URLLEN	Length of the contents of the URL header.
-	URLQUERY	Query portion of the contents of the URL header.
-	URLQUERYLEN	Length of the query portion of the URL header.
SSL	CLIENT.CERT	SSL client certificate as a whole.
-	CLIENT.CERT.SUBJECT	Contents of the client certificate subject field.
-	CLIENT.CERT.ISSUER	Client certificate issuer.
-	CLIENT.CERT.SIGALGO	Signature algorithm used in the client certificate.
-	CLIENT.CERT.VERSION	Client certificate version.
-	CLIENT.CERT.VALIDFROM	Date from which the client certificate is valid. (The start date.)
-	CLIENT.CERT.VALIDTO	Date after which the client certificate is no longer valid. (The end date.)
-	CLIENT.CERT.SERIALNUMBER	Client certificate serial number.
-	CLIENT.CIPHER.TYPE	Encryption method used in the client certificate.

Protocol	Qualifier	Definition
-	CLIENT.CIPHER.BITS	Number of significant bits in the encryption key.
-	CLIENT.SSL.VERSION	SSL version of the client certificate.
TCP	SOURCEPORT	Source port of the TCP connection.
-	DESTPORT	Destination port of the TCP connection.
-	MSS	Maximum segment size (MSS) of the TCP connection.
IP	SOURCEIP	Source IP address of the connection.
-	DESTIP	Destination IP address of the connection.

9. Choose the operator for your expression from the Operator drop-down list.

   Your choices depend on the qualifier you chose in the previous step. The complete list of operators that can appear in this drop-down list is:

   - == . Matches the following text string exactly.
   - != . Does not match the following text string.
     > - . Is greater than the following integer.
   - CONTAINS . Contains the following text string.
   - CONTENTS . The contents of the designated header, URL, or URL query.
   - EXISTS . The specified header or query exists.
   - NOTCONTAINS . Does not contain the following text string.
   - NOTEXISTS . The specified header or query does not exist.

   If you want this policy to operate on requests sent to a specific Host, you can leave the default, the equals (==) sign.

10. If the Value text box is visible, type the appropriate string or number into the text box.

    For example, if you want this policy to select requests sent to the host shopping.example.com, you would type that string in the Value text box.

11. If you chose HEADER as the qualifier, type the header you want in the Header Name text box.
12. Click OK to add your expression to the Expression list.
13. Repeat steps 4 through 11 to create additional expressions.
14. Click Close to close the Add Expression dialog box and return to the Create Cache Redirection Policy dialog box.

Cache redirection configurations

September 21, 2020

Depending on your deployment and network topology, you can configure one of the following types of cache redirection:

- **Transparent**. A transparent cache can reside on a variety of points along a network backbone to alleviate traffic along the delivery route. In transparent mode, the cache redirection virtual server intercepts all traffic flowing to the NetScaler appliance and applies cache redirection policies to determine whether content should be served from the cache or from the origin server.
- **Forward proxy**. A forward proxy cache server resides on the edge of an enterprise LAN and faces the WAN. In the forward proxy mode, the cache redirection virtual server resolves the hostname of the incoming request by using a DNS server and forwards requests for non-cacheable content to the resolved origin servers. Cacheable requests are sent to the configured cache servers.
- **Reverse proxy**. Reverse proxy caches are configured for specific origin servers. Incoming traffic directed to the reverse proxy, can either be served from a cache server or be sent to the origin server with or without modification to the URL.

Configure transparent redirection

September 21, 2020

When you configure transparent cache redirection, the NetScaler appliance evaluates all traffic it receives, to determine whether it is cacheable. This mode alleviates traffic along the delivery route and is often used when the cache server resides on the backbone of an ISP or carrier.

By default, cacheable requests are sent to a cache server, and non-cacheable requests to the origin server. For example, when the NetScaler appliance receives a request that is directed to a web server, it compares the HTTP headers in the request with a set of policy expressions. If the request does not match the policy, the appliance forwards the request to a cache server. If the request does match a policy, the appliance forwards the request, unchanged, to the web server.

For details on how to modify this default behavior, see “Direct policy hits to the cache instead of the origin.”

To configure transparent redirection, first enable cache redirection and load balancing, and configure edge mode. Then, create a cache redirection virtual server with a wildcard IP address (*), so that this virtual server can receive traffic coming to the appliance on any IP address the appliance owns. To this virtual server, bind cache redirection policies that describe the types of requests that should not be cached. Then, create a load balancing virtual server that will receive traffic from the cache redirection virtual server for cacheable requests. Finally, create a service that represents a physical cache server and bind it to the load balancing virtual server.

Enable cache redirection and load balancing

September 21, 2020

The appliance cache redirection and load balancing features are not enabled by default. They must be enabled before any cache redirection configuration can take effect.

Enable cache redirection and load balancing by using the CLI

At the command prompt, type the following command to enable cache redirection and load balancing and verify the settings:

```
-  enable ns feature cr lb
-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature cr lb
 Done
> show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   ON
 3)     Load Balancing                 LB                   ON
 4)     Content Switching              CS                   ON
 5)     Cache Redirection              CR                   ON
 6)     Sure Connect

```

```
            ...
            ...
            ...

 23)    HTML Injection                 HTMLInjection        ON
 24)    appliance Push                 push                 OF
 Done
<!--NeedCopy-->
```

Enable cache redirection and load balancing by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. To enable cache redirection, in the details pane, under Modes and Features, click Configure advanced features.
    a) In Configure Advanced Features dialog box, select the check box next to the Cache Redirection, and then click OK.
    b) In Enable/Disable Feature(s)? dialog box, click Yes.
3. To enable load balancing, in the details pane, under Modes and Features, click Configure basic features.
    a) In Configure Basic Features dialog box, select the check box next to the Load Balancing, and then click OK.
    b) In Enable/Disable Feature(s)? dialog box, click Yes.

Configure edge mode

September 21, 2020

When deployed at the edge of a network, the NetScaler appliance dynamically learns about the servers on that network. Edge mode enables the appliance to dynamically learn about up to 40,000 HTTP servers and proxy TCP connections for these servers.

This mode turns on the collection of statistics for the dynamically learned services and is typically used in transparent deployments for cache redirection.

Enable edge mode by using the CLI

At the command prompt, type the following commands to enable edge mode and verify the setting:

```
-  enable ns mode Edge
-  show ns mode
```

```
<!--NeedCopy-->
```

Example:

```
> enable ns mode edge
 Done

> show ns mode

        Mode                           Acronym             Status
        -------                        -------             ------
          ...
          ...
          ...
 6)     MAC-based forwarding           MBF                 ON
 7)     Edge configuration             Edge                ON
 8)     Use Subnet IP                  USNIP               OFF
          ...
          ...
          ...
 16)    Bridge BPDUs                   BridgeBPDUs         OFF
 Done
<!--NeedCopy-->
```

Enable edge mode by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure modes.
3. In Configure Modes dialog box, select the check box next to the Edge Configuration, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

Configure a cache redirection virtual server

September 21, 2020

By default, a cache redirection virtual server forwards cacheable requests to the load balancing virtual server for the cache, and forwards non-cacheable requests to the origin server (except in a reverse proxy configuration, in which non-cacheable requests are sent to a load balancing virtual server). There are three types of cache redirection virtual servers: transparent, forward proxy, and reverse proxy.

A transparent cache redirection virtual server uses an IP address of * and a port number, usually 80, that can accept HTTP traffic sent to any IP address that the appliance represents. As a result, you can configure only one transparent cache redirection virtual server. Any additional cache redirection virtual servers that you configure must be forward proxy or reverse proxy redirection servers.

add a cache redirection virtual server in transparent mode by using the cli

At the command prompt, type the following commands to add a cache redirection virtual server and verify the configuration:

```
-  add cr vserver <name> <serviceType> [<IPAddress> <port> ] [-cacheType <cacheType>] [-redirect <redirect>]
-  show cr vserver [<name>]
<!--NeedCopy-->
```

Example:

```
add cr vserver Vserver-CRD-1 HTTP * 80 -cacheType TRANSPARENT -redirect POLICY
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
 Done
<!--NeedCopy-->
```

Modify or remove a cache redirection virtual server by using the CLI

- To modify a virtual server, use the set cr vserver command, which is just like using the add cr vserver command, except that you enter the name of an existing virtual server.
- To remove a virtual server, use the `rm cr vserver` command, which accepts only the <name> argument.

Add a cache redirection virtual server in transparent mode by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.

2. In the details pane, click Add.
3. In the Create Virtual Server (Cache Redirection) dialog box, specify values for the following parameters as shown:
    - Name*—name
    - Port*—port

   *A required parameter
4. In the Protocol drop-down list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the standard port for the selected protocol, enter a new value in the Port field.
5. Click the Advanced tab.
6. Verify that Cache Type is set to TRANSPARENT and Redirect is set to POLICY.
7. Click Create, and then click Close. The Cache Redirection Virtual Servers pane displays the new virtual server.
8. Select the new cache redirection virtual server to display the details of its configuration.

Bind policies to the cache redirection virtual server

September 21, 2020

Cache redirection policies are not automatically bound to the cache redirection virtual server. A policy based cache redirection virtual server cannot function unless you bind at least one policy to it.

Bind policies to a cache redirection virtual server by using the CLI

At the command prompt, type:

```
-  bind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cache-control
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-dynamic-url
 Done
> bind cr vserver Vserver-CRD-1 -policyName bypass-urltokens
 Done
```

```
> bind cr vserver Vserver-CRD-1 -policyName bypass-cookie
 Done

> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache:
           TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: bypass-dynamic-url
3)      Cache bypass  Policy: bypass-urltokens
4)      Cache bypass  Policy: bypass-cookie
 Done
<!--NeedCopy-->
```

Bind a user-defined policy to a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and click Open.
3. On the Policies tab, select type of the policy and then click Insert Policy.
4. Under Policy Name column, select the policy that you want to bind.
5. Click OK.

Unbind a policy from a cache redirection virtual server

September 21, 2020

When you unbind a policy from the cache redirection virtual server, the NetScaler appliance no longer applies the policy when evaluating client requests.

Unbind a policy from a cache redirection virtual server by using the command CLI

At the command prompt, type:

```
-  unbind cr vserver <name> -policyName <string>
-  show cr vserver [<name>]
<!--NeedCopy-->
```

Example:

```
unbind cr vserver Vserver-CR-1 -policyName bypass-non-get
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP   ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: RULE        Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
 Done
<!--NeedCopy-->
```

Unbind a user-defined policy from a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. Click the virtual server that you want to configure, and then click Open.
3. On the Policies tab, under Policy Name, select the policy that you want to unbind.
4. Click Unbind Policy, and then click OK.

Create a load balancing virtual server

September 21, 2020

The cache redirection virtual server on the NetScaler appliance can send requests to either a cache server farm, if the request is cacheable, or to the origin server farm if the request is not cacheable.

Each cache server is represented on the appliance by a service, which is bound to a load balancing virtual server that receives requests from the cache redirection virtual server and forwards those requests to the servers.

For details on configuring load balancing virtual servers and other configuration options, see "Load Balancing."

Create a load balancing virtual server by using the CLI

At the command prompt, type the following commands to create a load balancing virtual server and verify the configuration:

```
-  add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> add lb vserver Vserver-LB-CR HTTP 10.102.20.30 80
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:00:08.470
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
 Done
<!--NeedCopy-->
```

Create a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
    - Name*-name
    - IP Address*- IPAddress

   - Port*-port

   *A required parameter

4. In the Protocol list, select a supported protocol (for example, **HTTP**). If the virtual server is to receive traffic on a port other than the well-known port for the selected protocol, enter a new value in the Port field.
5. Click Create, and then click Close. The Load Balancing Virtual Servers pane displays the new virtual server.

Configure an HTTP service

September 21, 2020

On the NetScaler appliance, a service represents a physical server on the network. In the transparent cache redirection configuration, the service represents the cache server. Cacheable requests are sent by the cache redirection virtual server to the load balancing virtual server, which in turn forwards each request to the correct service, which passes it on to the cache server.

Configure an HTTP service by using the CLI

At the command prompt, type the following commands to create an HTTP service and verify the configuration:

```
-  add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
-  show service [<name>]
<!--NeedCopy-->
```

Example:

```
> add service Service-HTTP-1 10.102.29.40 HTTP 80 -cacheType
   TRANSPARENT
 Done
> show service Service-HTTP-1
        Service-HTTP-1 (10.102.29.40:80) - HTTP
        State: DOWN
        Last state change was at Fri Jul  2 09:14:17 2010
        Time since last state change: 0 days, 00:00:13.820
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cache Type: TRANSPARENT Redirect Mode:
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1
                Probes: 3       Failed [Total: 3 Current: 3]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: N/A
 Done
<!--NeedCopy-->
```

Modify or remove a service by using the CLI

- To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service.
- To remove a service, use the `rm service` command, which accepts only the <name> argument.

Add an HTTP service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
    - Service Name*—name
    - Server*— IP
    - Port*—port

   *A required parameter
4. In the Protocol* drop-down list, select a supported protocol (for example, **HTTP**).
5. Click Create, and then click Close.

Bind/unbind a service to/from a load balancing virtual server

September 21, 2020

You must bind a service to the load balancing virtual server. This enables the load balancer to forward the request to the server that the service represents. If your configuration changes, you can unbind a service from the load balancing virtual server.

Bind a service to a load balancing virtual server by using the CLI

At the command prompt, type:

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind lb vserver vserver-LB-CR service-HTTP-1
 Done
> show lb vserver Vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 00:42:25.610
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

Unbind a service from a load balancing virtual server by using the CLI

To unbind a service, use the `unbind lb vserver` command instead of `bind lb vserver`.

Bind/unbind a service from a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server from which you want to bind/unbind the service, and then click Open.
3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

Disable the use the proxy port setting for transparent caching

September 21, 2020

If the use source IP (USIP) option is disabled on a cache service configured on the NetScaler appliance, the appliance forwards client requests to the cache service by using a appliance-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address and a random port as the source port. The randomly selected port is called the proxy port.

However, if you want to configure a fully transparent cache (a cache configuration in which the cache service receives the client's IP address and port number), you must not only enable the USIP option, either globally or on the cache service, but also disable the Use Proxy Port setting, either globally or on the cache service. Disabling the Use Proxy Port setting enables the appliance to use the client's source port as the source port when it connects to the cache service, and ensures a fully transparent cache configuration.

For more information about configuring the Use Proxy Port option globally or on a service, see Configuring the Source Port for Server-Side Connections.

Assign a port range to the NetScaler appliance

September 21, 2020

Sharing of the client IP address may create a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

A method to solve this problem is to assign a source port range to the NetScaler appliance. This allotment enables network devices to unambiguously identify the NetScaler appliance that sent the request.

Assign a source port range to a NetScaler appliance by using the CLI

At the command prompt, type:

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```

Assign a source port range to a NetScaler appliance by using the appliance GUI

1. In the navigation pane, click System, and then click Settings.
2. In the Settings group, click the Change global system settings link.
3. In the Cache Redirection Port Range group, specify the port range for the appliance by typing a port number for Start Port and a port number for End Port.
4. Click OK.

Enable load balancing virtual servers to redirect requests to cache

September 21, 2020

If a load balancing virtual server is configured to listen on a particular IP address and port combination, it takes precedence over the cache redirection virtual server for any requests destined for that address-port combination. Therefore, the cache redirection virtual server does not process those requests.

If you want to override this functionality and let the cache redirection virtual server decide whether the request should be served from the cache or not, configure the particular load balancing virtual server to be cacheable.

Such a configuration is typically used when an ISP uses a NetScaler appliance at the edge of its network and all traffic flows through the appliance.

Enable load balancing virtual servers to redirect requests to the cache by using the CLI

At the command prompt, type:

```
-  set lb vserver <name> [-cacheable ( YES | NO)]
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-CR  – cacheable YES
> show lb vserver vserver-LB-CR
        Vserver-LB-CR (10.102.20.30:80) - HTTP  Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 08:47:52 2010
        Time since last state change: 0 days, 01:05:51.510
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Cacheable: YES  PQ: OFF SC: OFF
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1) Service-HTTP-1 (10.102.29.40: 80) - HTTP State: DOWN Weight: 1
 Done
<!--NeedCopy-->
```

For transparent cache redirection, the appliance intercepts all traffic and evaluates every request to determine whether it is cacheable. Non-cacheable requests are sent unchanged to the origin server.

When using transparent cache redirection, you may want to turn off cache redirection for load balancing virtual servers that always direct traffic to origin servers.

Turn off caching for a load balancing virtual server by using the CLI

To turn off caching for a load balancing virtual, use the unset lb vserver command instead of set lb vserver. Specify a value of NO value for the **cacheable** parameter.

Enable or disable load balancing virtual servers to redirect requests to the cache by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server from which you want to enable/disable the caching, and then click Open.

3. On the Advanced tab, select/clear Cache Redirection check box.
4. Click OK.

Configure forward proxy redirection

September 21, 2020

A forward proxy is a single point of contact for a client or group of clients. In this configuration, the NetScaler appliance redirects non-cacheable requests to an origin server and redirects cacheable requests to either a forward proxy cache or a transparent cache.

When the appliance is configured as a forward proxy, users must modify their browsers so that the browser sends requests to the forward proxy instead of the destination servers.

A forward proxy cache redirection virtual server on the appliance compares the request with a policy for caching. If the request is not cacheable, the appliance queries a DNS load balancing virtual server for resolution of the destination, and then sends the request to the origin server. If the request is cacheable, the appliance forwards the request to a load balancing virtual server for the cache.

The appliance relies on a host domain name or IP address in the request's HOST header to determine the requested destination. If there is no HOST header in the request, the appliance inserts a HOST header based on the destination IP address in the request.

Typically, the NetScaler appliance acts as a forward proxy in an enterprise LAN. In such a configuration, the appliance resides at the edge of an enterprise LAN and intercepts client requests before they are fanned out to the WAN. Configuring the appliance in the forward proxy mode reduces traffic on the WAN.

To configure forward proxy cache redirection, first enable load balancing and cache redirection on the appliance. Then, configure a DNS load balancing virtual server and associated services. Also configure a load balancing virtual server and bind to it appropriate services for the cache. Configure a forward proxy cache redirection virtual server and bind the DNS and load balancing virtual servers to it. You must also configure caching policies and bind them to the cache redirection virtual server. To complete the setup, configure the client browsers to use the forward proxy.

For details on how to enable cache redirection and load balancing on the appliance, see "Enable cache redirection and load balancing."

For details on how to create a load balancing virtual server, see "Create a load balancing virtual server."

For details on how to configure services that represent the cache server, see "Configure an HTTP service."

For details on how to bind the service to a virtual server, see "Bind/unbind a service to/from a load balancing virtual server."

For details on how to create a forward proxy cache redirection server, see "Configure a cache redirection virtual server", and create a virtual server of type TRANSPARENT or FORWARD.

For details on binding cache redirection policies to the cache redirection virtual server, see "Configure a cache redirection policy."

Create a DNS service

September 21, 2020

A DNS service is a representation, on the NetScaler appliance, of a physical DNS server in the network. A DNS load balancing virtual server sends DNS requests to the DNS server in the network through such a service.

Create a DNS service by using the CLI

At the command line, type the following commands to create a DNS service and verify the configuration :

```
-  add service <name> <IP> <serviceType> <port>
-  show service [<name>]
<!--NeedCopy-->
```

Example:

```
add service Service-DNS-1 10.102.29.41 DNS 53
show service Service-DNS-1
        Service-DNS-1 (10.102.29.41:53) - DNS
        State: DOWN
        Last state change was at Fri Jul  2 10:14:32 2010
        Time since last state change: 0 days, 00:00:13.550
        Server Name: 10.102.29.41
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
```

```
            Client IP: DISABLED
            Cacheable: NO
            SC: OFF
            SP: OFF
            Down state flush: ENABLED

1)          Monitor Name: ping-default
                    State: DOWN     Weight: 1
                    Probes: 3       Failed [Total: 3 Current: 3]
                    Last response: Failure - Probe timed out.
                    Response Time: 2000.0 millisec
 Done
<!--NeedCopy-->
```

Add a DNS service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
    - Service Name*—name
    - Server*—IP
    - Port*—port

*A required parameter

1. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
2. Click Create, and then click Close.

Create a DNS load balancing virtual server

September 21, 2020

The DNS virtual server enables the forward proxy to perform DNS resolution before forwarding a client request to an origin server. The DNS load balancing virtual server is associated with the DNS service that represents the physical DNS server on the network.

Create a DNS load balancing virtual server by using the CLI

At the command line, type the following commands to create a DNS load balancing virtual server and verify the configuration:

```
-  add lb vserver <name> <serviceType>
-  show lb vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> add lb vserver Vserver-DNS-1 DNS
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:00:08.10
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
 Done
<!--NeedCopy-->
```

Create a DNS load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, in the Name box, type a name for the virtual server.
4. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
5. Click Create, and then click Close. The DNS Virtual Servers pane displays the new virtual server.

Bind the DNS service to the virtual server

September 21, 2020

For the DNS server to respond to DNS requests, the service representing the DNS server must be bound to the DNS virtual server.

Bind the DNS service to the load balancing virtual server by using the CLI

At the command prompt, type the following commands to bind the DNS service to the load balancing virtual server and verify the configuration:

```
-  bind lb vserver <name> <serviceName>
-  show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind lb vserver Vserver-DNS-1 Service-DNS-1
 Done
> show lb vserver Vserver-DNS-1
        Vserver-DNS-1 (0.0.0.0:0) - DNS Type: ADDRESS
        State: DOWN
        Last state change was at Fri Jul  2 10:32:28 2010
        Time since last state change: 0 days, 00:12:16.80
        Effective State: DOWN  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE

1) Service-DNS-1 (10.102.29.41: 53) - DNS State: DOWN   Weight: 1
 Done
>
<!--NeedCopy-->
```

Unbind a DNS service from the load balancing virtual server by using the CLI

Use the `unbind lb vserver` command instead of `bind lb vserver`.

Bind/Unbind a DNS service to/from a load balancing virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers
2. In the details pane, select the virtual server to/from which you want to bind/unbind the DNS service, and then click Open.
3. On the Services tab, in the Active column, select/clear the check box next to the Service Name.
4. Click OK.

Configure a client web browser to use a forward proxy

September 21, 2020

When you configure the NetScaler appliance as forward proxy cache redirection virtual server in the network, you must configure the client Web browser to send requests to the forward proxy. Typically, when you use a forward proxy, the only route to the servers in the network is through the forward proxy.

Refer the documentation for your browser to configure the browser to use a forward proxy. Specify the IP address and port number of the forward proxy cache redirection virtual server for this configuration.

Configure reverse proxy redirection

September 21, 2020

A reverse proxy resides in front of one or more Web servers and shields the origin server from client requests. Often, a reverse proxy cache is a front-end for all client requests to a server. An administrator assigns a reverse proxy cache to a specific origin server. This is unlike transparent and forward proxy caches, which cache frequently requested content for all requests to any origin server, and the choice of a server is based on the request.

Unlike a transparent proxy cache, the reverse proxy cache has its own IP address and can replace destination domains and URLs in a non-cacheable request with new destination domains and URLs.

You can deploy reverse proxy cache redirection at the origin-server side or at the edge of a network. When deployed at the origin server, the reverse proxy cache redirection virtual server is a front-end for all requests to the origin server.

In the reverse proxy mode, when the appliance receives a request, a cache redirection virtual server evaluates the request and forwards it to either a load balancing virtual server for the cache or a load balancing virtual server for the origin. The incoming request can be transformed by changing the host header or the host URL before they it is sent to the backend server.

To configure reverse proxy cache redirection, first enable cache redirection and load balancing. Then, configure a load balancing virtual server and services to send cacheable requests to the cache servers. Also configure a load balancing virtual server and associated services for the origin servers. Then, configure a reverse proxy cache redirection virtual server and bind relevant cache redirection policies to it. Finally, configure mapping policies and bind them to the reverse proxy cache redirection virtual server.

The mapping policies have an associated action that enables the cache redirection virtual server to forward any non-cacheable request to the load balancing virtual server for the origin.

Be sure to create the default cache server destination.

For details on how to enable cache redirection and load balancing on the appliance, see Enable cache redirection and load balancing.

For details on how to create a load balancing virtual server, see Create a load balancing virtual server.

For details on how to configure services that represent the cache server, see Configure an HTTP service.

For details on how to bind the service to a virtual server, see Bind/unbind a service to/from a load balancing virtual server.

For details on how to create a reverse proxy cache redirection server, see Configure a cache redirection virtual server, and create a virtual server of type REVERSE.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Bind policies to the cache redirection virtual server.

Configure mapping policies

If an incoming request is non-cacheable, the reverse-proxy cache redirection virtual server replaces the domain and URL in the request with the domain and URL of a target origin server and forwards the request to the load balancing virtual server for the origin.

A mapping policy enables the reverse proxy cache redirection virtual server to replace the destination domain and URL and forward the request to the load balancing virtual server for the origin.

A mapping policy must first translate the domain and the URL, and then pass the request on to the origin load balancing virtual server.

A mapping policy can map a domain, a URL prefix, and a URL suffix, as follows:

- Domain mapping: You can map a domain without a prefix or suffix. The domain mapping is the default mapping for the virtual server (for example, mapping www.mycompany.com to www.myrealcompany.com).
- Prefix mapping: You can replace a specified pattern prefixed as part of the URL (for example, mapping www.mycompany.com/sports/index.html to www.mycompany.com/news/index.html).
- Suffix mapping: You can replace the file suffix in the URL (for example, mapping www.mycompany.com/sport to www.mycompany.com/sports/index.asp).

The source and the destination strings being mapped must be similar. If you specify a source domain, you must specify a destination domain, and if you specify a source suffix, you must specify a destination suffix. Similarly, if you specify an exact URL from the source, the target URL must also be an exact URL.

Once you configure mapping policies for the reverse proxy mode, you must bind them to the cache redirection virtual server.

You can use combinations of the source URL, target URL, and source and target domains to configure all three types of domain mapping.

Configure a mapping policy for reverse proxy mode by using the CLI

At the command prompt, type the following command to add a policy map and verify the configuration:

```
-  add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <string>] [-tu <string>]
-  show policy map [<mapPolicyName>]
<!--NeedCopy-->
```

Example:

The following command maps a domain in a client request to a target domain:

```
> add policy map myMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com
 Done
> show policy map myMappingPolicy
1)      Name: myMappingPolicy
        Source Domain: www.mycompany.com        Source Url:
        Target Domain: www.myrealcompany.com    Target Url:
 Done
<!--NeedCopy-->
```

Following is an example of mapping a URL suffix to a different URL suffix:

```
> add policy map myOtherMappingPolicy -sd www.mycompany.com -td www.myrealcompany.com -su /news.html -tu /realnews.html
 Done
> show policy map myOtherMappingPolicy
1)      Name: myOtherMappingPolicy
        Source Domain: www.mycompany.com        Source Url: /news.html
        Target Domain: www.myrealcompany.com    Target Url: /realnews.html
 Done
<!--NeedCopy-->
```

Configure a mapping policy for reverse proxy mode by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Map Policies.
2. In the details pane, click Add.
3. In the Create Map Policy dialog box, specify values for the following parameters as shown:
    - Name*- mapPolicyName
    - Source Domain*-sd
    - Target Domain*-td
    - Source URL-su
    - Target URL-tu

    *A required parameter
4. Click Create, and then click Close. The Map pane displays the new mapping policy.

Bind the mapping policy to the cache redirection virtual server by using the CLI

At the command prompt, type the following commands to bind the mapping policy to the cache redirection virtual server and verify the configuration:

```
-  bind cr vserver <name> -policyName <string> [<targetVserver>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> bind cr vserver Vserver-CRD-3 -policyName myMappingPolicy Vserver-LB-CR
 Done
> show cr vserver Vserver-CRD-3
        Vserver-CRD-3 (10.102.29.50:88) - HTTP  Type: CONTENT
        State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default: Vserver-LB-CR  Content Precedence: RULE        Cache: REVERSE
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Policy:         Target: Vserver-LB-CR   Priority: 0     Hits: 0
1)      Map: myMappingPolicy Target: Vserver-LB-CR
 Done
<!--NeedCopy-->
```

Bind the mapping policy to the cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server from which you want to bind the mapping policy, and then click Open.
3. In the Configure Virtual Server(Cache Redirection), on the Policies tab, select Map, and then click Insert Policy.
4. In the Policy Name column, select the policy from drop down list.
5. In the Target column, click the down arrow, and then select the vserver from drop down list.
6. Click OK.

Selective cache redirection

September 21, 2020

Selective cache redirection sends requests for particular types of content, for example, images, to one cache server or group of cache servers and sends other types of content to a different cache server or group of cache servers. You can configure advanced cache redirection in transparent, reverse proxy, or forward proxy modes.

In selective cache redirection, the NetScaler appliance intercepts a client request and forwards noncacheable requests to the original destination in the client request. For cacheable requests, the appliance sends the requests to the destination cache server that can serve content of a specific content type.

Selective cache redirection involves configuring content switching policies in addition to cache redirection policies. The appliance first evaluates the cache redirection policies that are bound to the cache redirection virtual server. If a request matches a cache redirection policy, the cache redirection virtual server sends the request to the origin server or a load balancing virtual server for the origin. If no cache redirection policies match the request, the appliance evaluates the content switching policies bound to the cache redirection virtual server. If a content switching policy matches the request, the cache redirection virtual server redirects the request to a load balancing virtual server for the cache.

To configure selective cache redirection, first enable cache redirection, load balancing, and content switching on the NetScaler appliance. Then, configure a load balancing virtual server for the cache and an associated HTTP service. After this, configure a cache redirection virtual server and bind both the cache redirection and content switching policies to it. Once you have bound the policies, you can configure the virtual server to give precedence to either rule based or URL based content-switching policies.

When configured for transparent mode cache redirection in an edge deployment topology, the appliance sends all cacheable HTTP traffic to a transparent cache farm. Clients access the Internet through the appliance, which is configured as a Layer 4 switch that receives traffic on port 80.

The appliance can direct requests for images (for example, .png and .jpg files) to one server in the transparent cache farm, and all other requests for static content to other servers in the farm. For this configuration, you configure content switching policies to send images to the image cache and send all other cacheable content to a default cache.

> **Note**
>
> The configuration described here is for transparent selective cache redirection. Therefore, it does not require a load balancing virtual server for the origin, as would a reverse proxy configuration.

To configure this type of selective cache redirection, first enable cache redirection, load balancing, and content switching. Then, configure a load balancing virtual server for the cache and configure an associated HTTP service. Then, configure a cache redirection virtual server and create and bind both cache redirection and content switching policies to this virtual server.

For details on how to enable cache redirection and load balancing on the appliance, see Enable cache redirection and load balancing.

Enable content switching

September 21, 2020

To configure selective cache redirection, after you enable both the load balancing and cache redirection features on the appliance, you must enable content switching.

Enable content switching by using the CLI

At the command prompt, type:

```
-  enable ns feature CS

-  show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature cs
 Done
> show ns feature

```

```
        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   ON
 3)     Load Balancing                 LB                   ON
 4)     Content Switching              CS                   ON
 5)     Cache Redirection              CR                   ON
          ...
          ...
          ...
 23)    HTML Injection                 HTMLInjection        ON
 24)    appliance Push                 push                 OFF
 Done
<!--NeedCopy-->
```

Enable cache redirection and load balancing by using the GUI

1. In the navigation pane, expand System, and then click Settings.
2. In the details pane, under Modes and Features, click Configure basic features.
3. In Configure Basic Features dialog box, select the check box next to the Content Switching, and then click OK.
4. In Enable/Disable Feature(s)? dialog box, click Yes.

Configure a load balancing virtual server for the cache

September 21, 2020

Create a load balancing virtual server and an HTTP service for each type of cache server that will be used. For example, if you want to serve JPEG files from one cache server and GIF files from another cache server, and use a third cache server for the rest of the content, create an HTTP service and virtual server for each of the three types of cache servers. Then bind each service to its respective virtual server.

For details on how to create a load balancing virtual server, see Create a load balancing virtual server.

For details on how to configure services that represent the cache server, see Configure an HTTP service.

For details on how to bind the service to a virtual server, see Bind/unbind a service to/from a load balancing virtual server.

For details on how to create a transparent proxy cache redirection server, see Configure a cache redirection virtual server, and create a virtual server of type TRANSPARENT.

For details on binding built-in cache redirection policies to the cache redirection virtual server, see Bind policies to the cache redirection virtual server.

Configure a cache redirection policy for a specific type of content

To identify requests that contain a .png or .jpeg extension as cacheable, you configure a cache redirection policy and bind it to the cache redirection virtual server.

> **Note**
>
> If a request matches a policy, the NetScaler appliance forwards it to the origin server. As a result, in the following procedure, you configure policies to match requests that do
> *not* have ".png" or ".jpeg" extensions.

To configure cache redirection for a specific type of content, configure a policy that uses a simple expression, as described in Configure a cache redirection policy.

Configure policies for content switching

September 21, 2020

You must create a content switching policy to identify specific types of content to be cached in one cache server or farm and identify other types of content to serve from another cache server or farm. For example, you can configure a policy to determine the location for image files with .png and .jpeg extensions.

After defining the content switching policy, you bind it to a cache redirection virtual server and specify a load balancing virtual server. Requests that match the policy are forwarded to the named load balancing virtual server. Requests that do not match the content switching policy are forwarded to the default load balancing virtual server for the cache.

For more details about the content switching feature and configuring content switching policies, see Content switching.

You must first create the content switching policy and then bind it to the cache redirection virtual server.

Create a content switching policy by using the command CLI

At the command line, type:

```
-  add cs policy <policyName> [-url <string> | -rule <expression>]
-  show cs policy [<policyName>]
```

```
<!--NeedCopy-->
```

Examples:

```
> add cs policy Policy-CS-JPEG -rule "REQ.HTTP.URL == '/*.jpeg'"
 Done
> show cs policy Policy-CS-JPEG
        Rule: REQ.HTTP.URL == '/*.jpeg'         Policy: Policy-CS-JPEG
        Hits: 0
 Done
>

> add cs policy Policy-CS-GIF -rule "REQ.HTTP.URL == '/ *.png'"
 Done
> show cs policy Policy-CS-GIF
        Rule: REQ.HTTP.URL == '/ *.png'         Policy: Policy-CS-GIF
        Hits: 0
 Done
>

> add cs policy Policy-CS-JPEG-URL -url /*.jpg
 Done
> show cs policy Policy-CS-JPEG-URL
        URL: /*.jpg     Policy: Policy-CS-JPEG-URL
        Hits: 0
 Done
>

> add cs policy Policy-CS-GIF-URL -url /*.png
 Done
> show cs policy Policy-CS-GIF-URL
        URL: /*.png     Policy: Policy-CS-GIF-URL
        Hits: 0
 Done
<!--NeedCopy-->
```

Create a URL-based content switching policy by using the GUI

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the URL radio button.

5. In the Value text box, type the string value (for example, **/sports**).
6. Click Create and click Close. The policy you created appears in the Content Switching Policies page.

Create a rule-based content switching policy by using the GUI

1. Navigate to Traffic Management > Content Switching > Policies.
2. In the details pane, click Add.
3. In the Create Content Switching Policy dialog box, in the Name text box, type a name for the policy.
4. Select the Expression radio button, and then click Configure.
5. In the Create Expression dialog box, choose the expression syntax that you want to use.

    - If you want to use default syntax, accept the default and proceed to the next step.
    - If you want to use classic syntax, click Switch to Classic Syntax.

    The Expression portion of the dialog box changes to match your choice. The default syntax Expression view has fewer elements than does the classic syntax Expression view. In the default syntax Expression view, instead of a preview window, a button provides access to an expression evaluator. The evaluator evaluates the expression you entered, to verify that it is valid, and displays an analysis of the expression's effect.

6. Enter your policy expressions.

    For information about using the advanced syntax, see Configure advanced policy expression: Get started.

7. Click Create and click Close. The policy you created appears in the Content Switching Policies pane.

Bind the content switching policy to a cache redirection virtual server by using the CLI

At the command prompt, type the following commands to bind the content switching policy to a cache redirection virtual server and verify the configuration:

```
-  bind cs vserver <name> <targetVserver> [-policyName <string>]
-  show cs vserver [<name>]
<!--NeedCopy-->
```

Example:

```
> bind cs vserver Vserver-CR-1 lbcachejpeg -policyName Policy-CS-JPEG
 Done
```

```
> bind cs vserver Vserver-CR-1 lbcachegif -policyName Policy-CS-GIF
 Done
> show cs vserver Vserver-CR-1
        Vserver-CR-1 (10.102.29.60:80) - HTTP   Type: CONTENT
        State: UP
        Last state change was at Fri Jul  2 12:53:45 2010
        Time since last state change: 0 days, 00:00:58.920
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Cacheable: YES
        Vserver IP and Port insertion: OFF
        Case Sensitivity: ON
        Push: DISABLED  Push VServer:
        Push Label Rule: none

1)      Policy: Policy-CS-JPEG  Target: lbcachejpeg     Priority: 0     Hits: 0
2)      Policy: Policy-CS-GIF   Target: lbcachegif      Priority: 0     Hits: 0
 Done
>
<!--NeedCopy-->
```

Bind the content switching policy to a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server for which you want to bind the policy (for example, **Vserver-CS-1**), and then click Open.
3. In the Configure Virtual Server (Content Switching) dialog box, on the Policies tab, click CSW, and then click Insert Policy.
4. In the Policy Name column, select the policy that you want to configure for the content switching virtual server.
5. In the Target column, click the green arrow, and select the target load balancing virtual server from the list.
6. Click OK.

Configure precedence for policy evaluation

September 21, 2020

You can configure a content switching policy based on either a rule, which is a generic configuration to accommodate various content types, or a URL, which is more specific and defines exactly the type of content that has to be sent to a particular cache server. Essentially, the same content can be defined by either a rule based policy or a URL based policy.

Once you bind content switching policies of either type to a cache redirection virtual server, you can configure the virtual server to give precedence to either rule based or URL based policies. This will, in turn, decide which servers the particular requests are directed to.

To configure precedence for policy evaluation, use the precedence parameter, which specifies the type of policy (URL or RULE) that takes precedence on the content redirection virtual server.

Possible values: RULE, URL

Default value: RULE

Configure precedence for policy evaluation by using the CLI

At the command prompt, type the following commands to configure precedence for policy evaluation and verify the configuration:

```
-  set cr vserver <name> [-precedence (RULE | URL)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -precedence URL
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP   ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
```

```
 Done
>
<!--NeedCopy-->
```

Configure precedence for policy evaluation by using the GUI

1. Navigate to Traffic Management > Content Switching > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure precedence, (for example, **Vserver-CS-1**), and then click Open.
3. In the Configure Virtual Server (Content Switching) dialog box, on the Advanced tab, next to Precedence, click Rule or URL, and then click OK.

Administer a cache redirection virtual server

September 21, 2020

To administer a cache redirection virtual server, you need to view cache redirection statistics. You might need to enable or disable cache redirection servers, or direct policy hits to the cache instead of the origin. Administrative tasks also include backing up a cache redirection virtual server and managing client connections.

View cache redirection virtual server statistics

September 21, 2020

You can view properties of a cache redirection virtual server and statistics on the traffic that has passed through a cache redirection virtual server. You can also view the cache redirection virtual servers and policies that you have bound to load balancing virtual servers.

To view statistics for a specific cache redirection virtual servers, use the name parameter to specify the name of the virtual server for which statistics will be displayed. Otherwise, statistics for all cache redirection virtual servers are displayed. Maximum Length: 127

View statistics for a cache redirection virtual server by using the CLI

At the command prompt, type:

`stat cr vserver [<name>]`

Example:

```
> stat cr vserver Vserver-CRD-1

Vserver Summary
                          IP  port     Protocol        State
Vser...CRD-1         0.0.0.0    80         HTTP           UP

VServer Stats:
                                          Rate (/s)
                                             Total
Requests                                            0
                      0
Responses                                           0
                      0
Request bytes                                       0
                      0
Response bytes                                      0
                      0

 Done
>
<!--NeedCopy-->
```

View statistics for a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers
2. In the details pane, select the virtual server for which you want to view statistics, (for example, **Vserver-CRD-1**), and then click Statistics.

Omit the server name to display basic statistics for all cache redirection virtual servers. Include the server name to display detailed statistics for that virtual server, including number and size of requests and responses that pass through the virtual server

View the statistics of a cache redirection virtual server by using the monitoring and dashboard utilities

1. To view the statistics by using the monitoring utilities, click the Monitoring tab.
2. In the Select Group drop-down menu, choose CR Virtual Servers. A list of cache redirection virtual servers appears.
3. To view the statistics by using the dashboard utilities, click the Dashboard tab.
4. Click Applet Client or Web Start Client next to Statistical Utility.
5. In the Select Group drop-down menu, choose CR Virtual Servers. The dashboard displays summary statistics for the cache redirection virtual servers.

6. To see a chart of virtual server activity, click Chart. A graphical representation of the virtual server statistics appears.

Enable or disable a cache redirection virtual server

September 21, 2020

When you create a cache redirection virtual server, it is enabled by default. If you disable a cache redirection virtual server, its state changes to OUT OF SERVICE and it stops redirecting cacheable client requests. However, the NetScaler appliance continues to respond to ARP and ping requests for the IP address of this virtual server.

Enable or disable a cache redirection virtual server by using the CLI

At the command line, type one of the following commands:

```
-  enable cr vserver <name>
-  show cr vserver <name>
-  disable cr vserver <name>
-  show cr vserver <name>
<!--NeedCopy-->
```

Examples:

```
> enable cr vserver Vserver-CRD-1
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>

> disable cr vserver Vserver-CRD-1
```

```
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: OUT OF SERVICE  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: ORIGIN L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

Enable or disable a cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the navigation pane, expand Cache Redirection, and then click Virtual Servers.
3. In the details pane, select the virtual server that you want to enable or disable, (for example, **Vserver-CRD-1**), and then click Statistics.
4. In the Proceed dialog box, click Yes.

Direct policy hits to the cache instead of the origin

September 21, 2020

By default, when a request matches a policy, the NetScaler appliance forwards the request either to the origin server directly, or to a load balancing virtual server for the origin, depending on how you have configured cache redirection.

You can change the default behavior so that when a request matches a policy, the request is forwarded to a load balancing virtual server for the cache.

To change the destination for a policy hit to the origin or the cache, use the onPolicyMatch parameter, which specifies where to send requests that match the cache redirection policy.

The valid options are:

1. CACHE - Directs all matching requests to the cache.
2. ORIGIN - Directs all matching requests to the origin server.

> **Note**
>
> For this option to work, you must select the cacheredirection type as POLICY.

Possible values: CACHE, ORIGIN

Default value: ORIGIN

Change the destination for a policy hit to the origin or the cache by using the CLI

At the command prompt, type the following commands to change the destination for a policy hit and verify the configuration:

```
-  set cr vserver <name> [-onPolicyMatch (ORIGIN | CACHE)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -onPolicyMatch CACHE
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP   ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:         Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Change the destination for a policy hit to the origin or the cache by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, click Advanced tab.
4. Select CACHE or ORIGIN from the Redirect To drop-down list.
5. Click OK.

Back up a cache redirection virtual server

September 21, 2020

Cache redirection can fail if the primary virtual server fails, or if it is unable to handle excessive traffic. You can specify a backup virtual server to take over the processing of traffic when the primary virtual server fails.

To specify a backup cache redirection virtual server, use the backupVServer parameter, which specifies Backup Virtual Server. Maximum Length: 127

Specify a backup cache redirection virtual server by using the CLI

At the command prompt, type the following commands to specify a backup cache redirection virtual server and verify the configuration:

```
-  set cr vserver <name> [-backupVServer <string>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -backupVServer Vserver-CRD-2
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Specify a backup cache redirection virtual server by using the GUI

1. Navigate to Traffic Management > Cache Redirection > irtual Servers.

2. In the details pane, select the virtual server for which you want to change the destination for a policy hit, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Backup Virtual Server drop-down list, select the virtual server.
5. Click OK.

Manage client connections for a virtual server

September 21, 2020

You can configure timeouts on a cache redirection virtual server so that client connections are not kept open indefinitely. You can also insert Via headers in requests. To possibly reduce network congestion, you can reuse open TCP connections. You can enable or disable delayed cleanup of cache redirection virtual server connections.

You can configure the appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.

The following settings can be made on a virtual server:

- When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, appliance always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, appliance responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, appliance responds even if one virtual server set to ACTIVE is UP.

This document includes the following information:

- Configure client timeout
- Insert Via headers in the requests
- Reuse TCP connections
- Configure delayed connection cleanup

Configure client timeout

You can specify expiration of client requests by setting a timeout value for the cache redirection virtual server. The timeout value is the number of seconds for which the cache redirection virtual server waits to receive a response for the client request.

To configure a time-out value, use the cltTimeout parameter, which specifies the time, in seconds, after which the NetScaler appliance closes any idle client connections. The default value is 180sec for HTTP/SSL-based services and 9000sec for TCP-based services.

Configure client timeout by using the CLI

At the command prompt, type the following commands to configure client timeout and verify the configuration:

```
-  set cr vserver <name> [-cltTimeout <secs>]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -cltTimeout 6000
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP   ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Configure client timeout by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. In the Client Time-out(secs) text box, enter the time-out value in seconds.
5. Click OK.

Insert Via headers in the requests

A Via header lists the protocols and recipients between the start and end points for a request or a response and informs the server of proxies through which the request was sent. You can configure the cache redirection virtual server to insert a Via header in each HTTP request. The via parameter is enabled by default when you create a cache redirection virtual server.

To enable or disable Via-header insertion in client requests, use the via parameter, which specifies the state of the system in inserting a Via header in the HTTP requests.

Possible values: ON, OFF

Default value: ON

Enable or disable Via-header insertion in client requests by using the CLI

At the command prompt, type:

```
-  set cr vserver <name> [-via (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -via ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP   ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
>
<!--NeedCopy-->
```

Enable or disable Via-header insertion in client requests by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Via check box.
5. Click OK.

Reuse TCP connections

You can configure the NetScaler appliance to reuse TCP connections to the cache and origin servers across client connections. This can improve performance by saving the time required to establish a session between the server and the appliance. The reuse option is enabled by default when you create a cache redirection virtual server.

To enable or disable the reuse of TCP connections, use the reuse parameter, which specifies the state of reuse of TCP connections to the cache or origin servers across client connections.

Possible values: ON, OFF

Default value: ON

Enable or disable the reuse of TCP connections by using the CLI

At the command prompt, type:

```
-  set cr vserver <name> [-reuse (ON|OFF)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -reuse ON
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2
```

```

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Enable or disable the reuse of TCP connections by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, select the Advanced tab.
4. Select the Reuse check box.
5. Click OK.

Configure delayed connection cleanup

The down state flush option performs delayed cleanup of connections on a cache redirection virtual server. The down state flush option is enabled by default when you create a cache redirection virtual server.

To enable or disable the down state flush option, set the downStateFlush parameter.

Possible values: ENABLED, DISABLED

Default value: ENABLED

Enable of disable the down state flush option by using the CLI

At the command prompt, type the following commands to configure delayed connection clean up and verify the configuration:

```
-  set cr vserver <name> [-downStateFlush (ENABLED | DISABLED)]
-  show cr vserver <name>
<!--NeedCopy-->
```

Example:

```
> set cr vserver Vserver-CRD-1 -downStateFlush ENABLED
 Done
> show cr vserver Vserver-CRD-1
        Vserver-CRD-1 (*:80) - HTTP     Type: CONTENT
        State: UP  ARP:DISABLED
        Client Idle Timeout: 6000 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Default:        Content Precedence: URL Cache: TRANSPARENT
        On Policy Match: CACHE  L2Conn: OFF     OriginUSIP: OFF
        Redirect: POLICY        Reuse: ON       Via: ON ARP: OFF
        Backup: Vserver-CRD-2

1)      Cache bypass  Policy: bypass-cache-control
2)      Cache bypass  Policy: Policy-CRD
 Done
<!--NeedCopy-->
```

Enable or disable the reuse of TCP connections by using the GUI

1. Navigate to Traffic Management > Cache Redirection > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure client timeout, (for example, **Vserver-CRD-1**), and then click Open.
3. In Configure Virtual Server (Cache Redirection) dialog box, click Advanced tab.
4. Select the Down state flush check box.
5. Click OK.

N-Tier cache redirection

September 21, 2020

To efficiently handle large amounts of cached data, typically several gigabytes per second, an Internet Service Provider (ISP) deploys several dedicated cache servers. The cache redirection feature of the NetScaler appliance can help load balance the cache servers, but a single appliance or a couple of appliances might not efficiently handle the large volume of traffic.

You can solve the problem by deploying the NetScaler appliances in two tiers (layers), where the appliances in the upper tier load balance those in the lower tier and the appliances in the lower tier load balance the cache servers. This arrangement is called *n-tier cache redirection*.

For purposes such as auditing and security, an ISP has to track client details such as the IP address, information provided, and the time of the interaction. Therefore, client connections through a NetScaler appliance have to be fully transparent. However, if you configure transparent cache redirection, with the NetScaler appliances deployed in parallel, the IP address of the client has to be shared among all the appliances. Sharing of the client IP address creates a conflict that makes network devices, such as routers, cache servers, origin servers, and other NetScaler appliances, unable to determine the appliance, and therefore the client, to which the response should be sent.

How N-tier cache redirection Is implemented

To solve the problem, appliance n-tier cache redirection splits the source port range among the appliances in the lower tier and includes the client IP address in the request sent to the cache servers. The upper-tier NetScaler appliances are configured to do sessionless load balancing in order to avoid unnecessary load on the appliances.

When the lower-tier NetScaler appliance communicates with a cache server, it uses a mapped IP address (MIP) to represent the source IP address. Therefore, the cache server can identify the appliance from which it received the request and send the response to the same appliance.

The lower-tier NetScaler appliance inserts the client IP address into the header of the request sent to the cache server. The client IP in the header helps the appliance to determine the client to which the packet should be forwarded when it receives the response from a cache server, or the origin server in case of a cache miss. The origin server determines the response to be sent according to the client IP inserted in the request header.

The origin server sends the response to an upper-tier appliance, including the source port number from which the origin server received the request. The entire source port range, 1024 to 65535, is distributed among the lower-tier NetScaler appliances. Each lower-tier appliance is exclusively assigned a group of addresses within the range. This allotment enables the upper-tier appliance to unambiguously identify the lower-tier NetScaler appliance that sent the request to the origin server. The upper-tier appliance can therefore forward the response to the correct lower-tier appliance.

The upper-tier NetScaler appliances are configured to do policy-based routing, and the routing policies are defined to determine the IP address of the destination appliance from the source port range.

Setup necessary for configuring N-Tier CRD

The following setup is necessary for the functioning of n-tier cache redirection:

For each upper-tier NetScaler appliance:

- Enable Layer 3 mode.
- Define policies for policy-based routes (PBRs) so that traffic is forwarded according to the range of the destination port.
- Configure a load balancing virtual server.
- Configure the virtual server to listen to all the traffic coming from the client. Set the Service Type/Protocol to be ANY and IP Address as asterisk (*).
- Enable sessionless load balancing with MAC-based redirection mode to avoid unnecessary load on the upper-tier NetScaler appliances.
- Make sure that the Use Proxy Port option is enabled.
- Create a service for each lower-tier appliance and bind all the services to the virtual server.

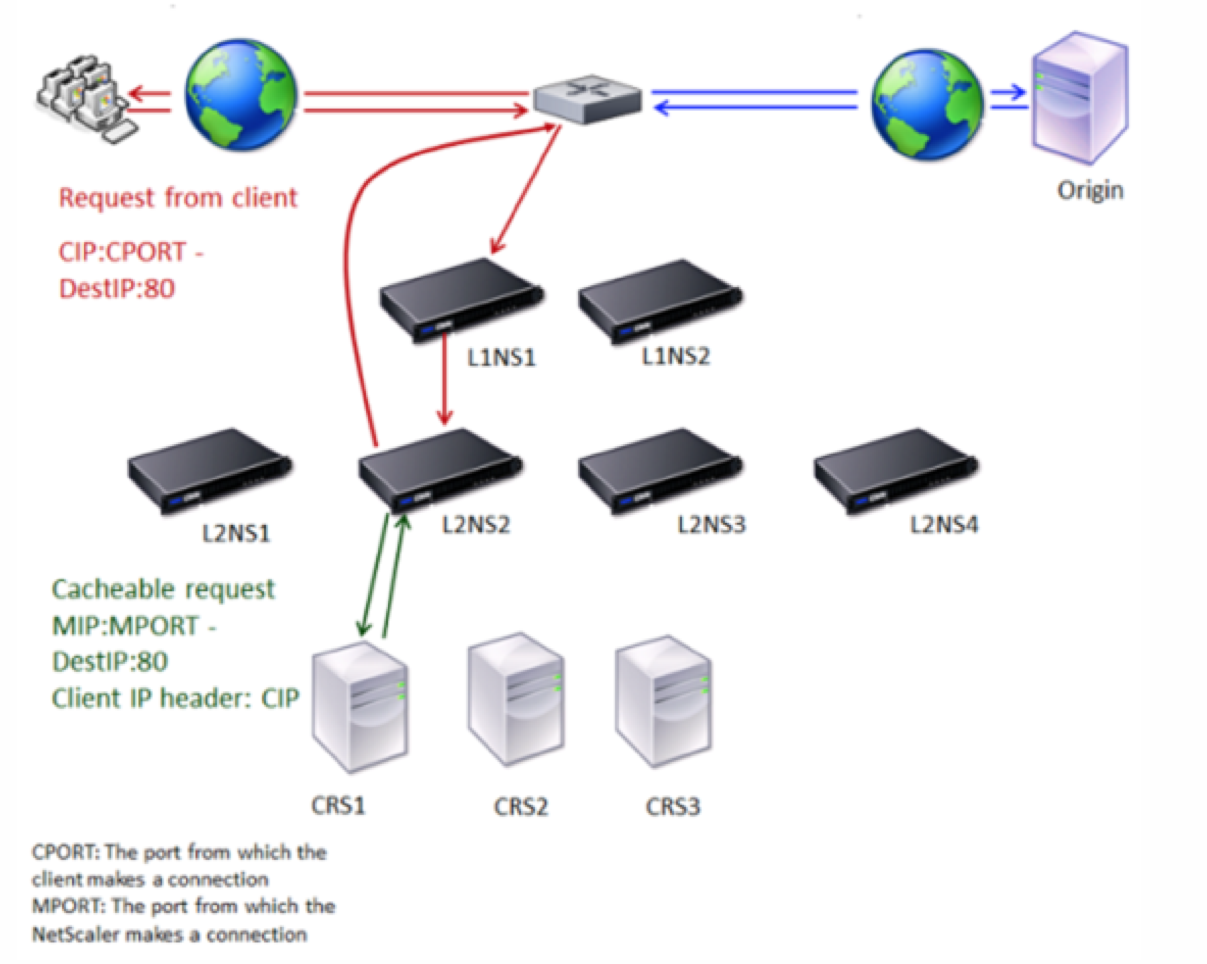
For each lower-tier NetScaler appliance,

- Configure the cache redirection port range on the appliance. Assign an exclusive range to each lower-tier appliance.
- Configure a load balancing virtual server and enable MAC-based redirection.
- Create a service for each cache server that is to be load balanced by this appliance. When creating the service, enable insertion of client IP in the header. Then, bind all the services to the load balancing virtual server.
- Configure a transparent mode cache redirection virtual server with the following settings:
  - Enable the Origin USIP option.
  - Add a source IP expression to include the client IP in the header.
  - Enable the Use Port Range option.

How N-tier cache redirection works during a cache hit

The following figure shows how cache redirection works when a client request is cacheable and the response is sent from a cache server.

Figure 1. Cache Redirection in Case of a Cache Hit


Origin
Request from client
CIP:CPORT -
DestIP:80
L1NS1
L1NS2
L2NS1
L2NS2
L2NS3
L2NS4
Cacheable request
MIP:MPORT -
DestIP:80
Client IP header: CIP
CRS1
CRS2
CRS3
CPORT: The port from which the client makes a connection
MPORT: The port from which the NetScaler makes a connection

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

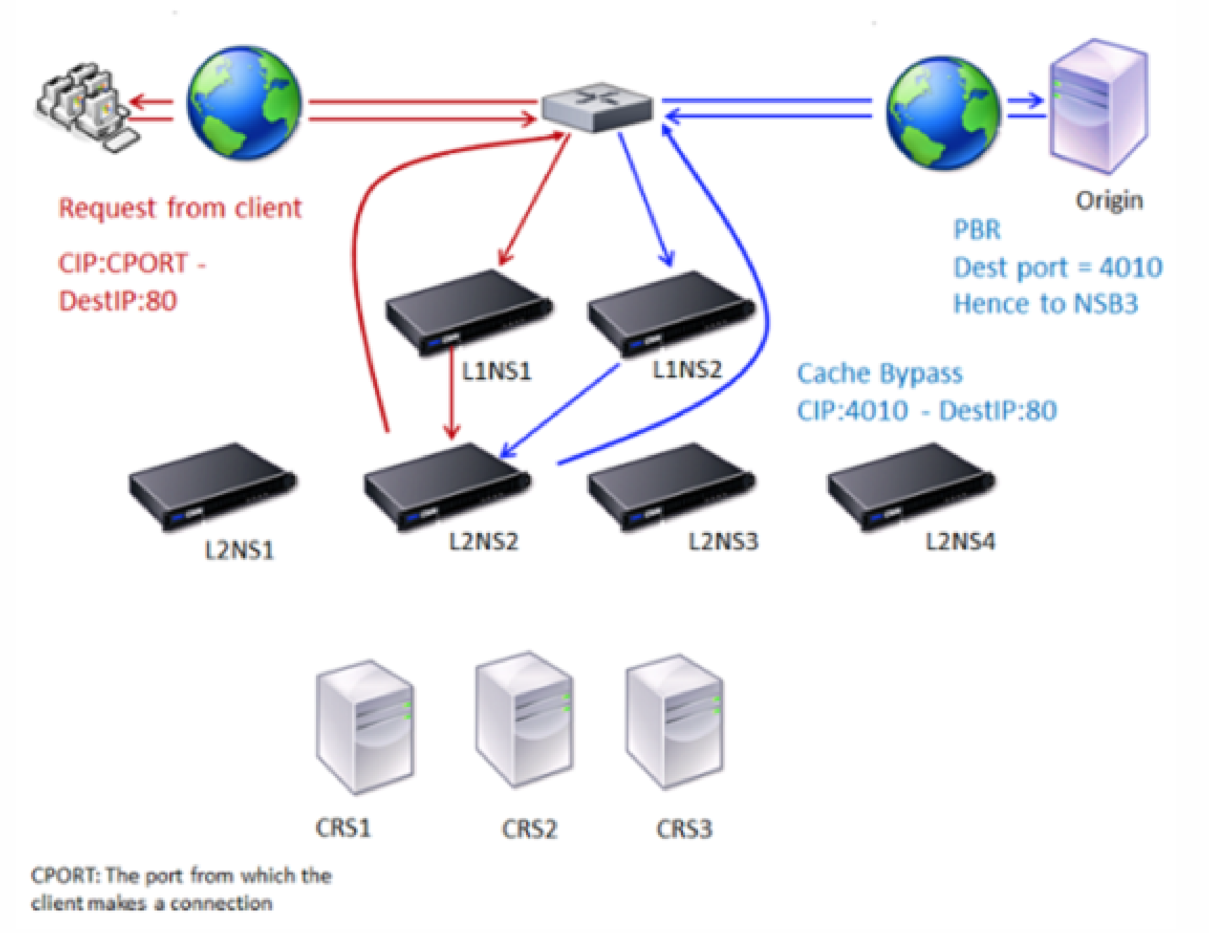
Traffic flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1, and the request is cacheable. L2NS2 includes the client IP in the request header.
4. CRS1 sends the response to L2NS2 because L2NS2 used its MIP as the source IP address when connecting to CRS1.
5. With the help of the client IP address in the request header, L2NS2 identifies the client from which the request came. L2NS2 directly sends the response to the router, avoiding unnecessary load on the appliance in the upper tier.
6. The router forwards the response to Client A.

How N-tier cache redirection works during a cache bypass

The following figure shows how cache redirection works when a client request is sent to an origin server for a response.

Figure 2. Cache Redirection in Case of a Cache Bypass


Request from client
CIP:CPORT -
DestIP:80
Origin
PBR
Dest port = 4010
Hence to NSB3
L1NS1
L1NS2
Cache Bypass
CIP:4010 - DestIP:80
L2NS1
L2NS2
L2NS3
L2NS4
CRS1
CRS2
CRS3
CPORT: The port from which the
client makes a connection

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic flow

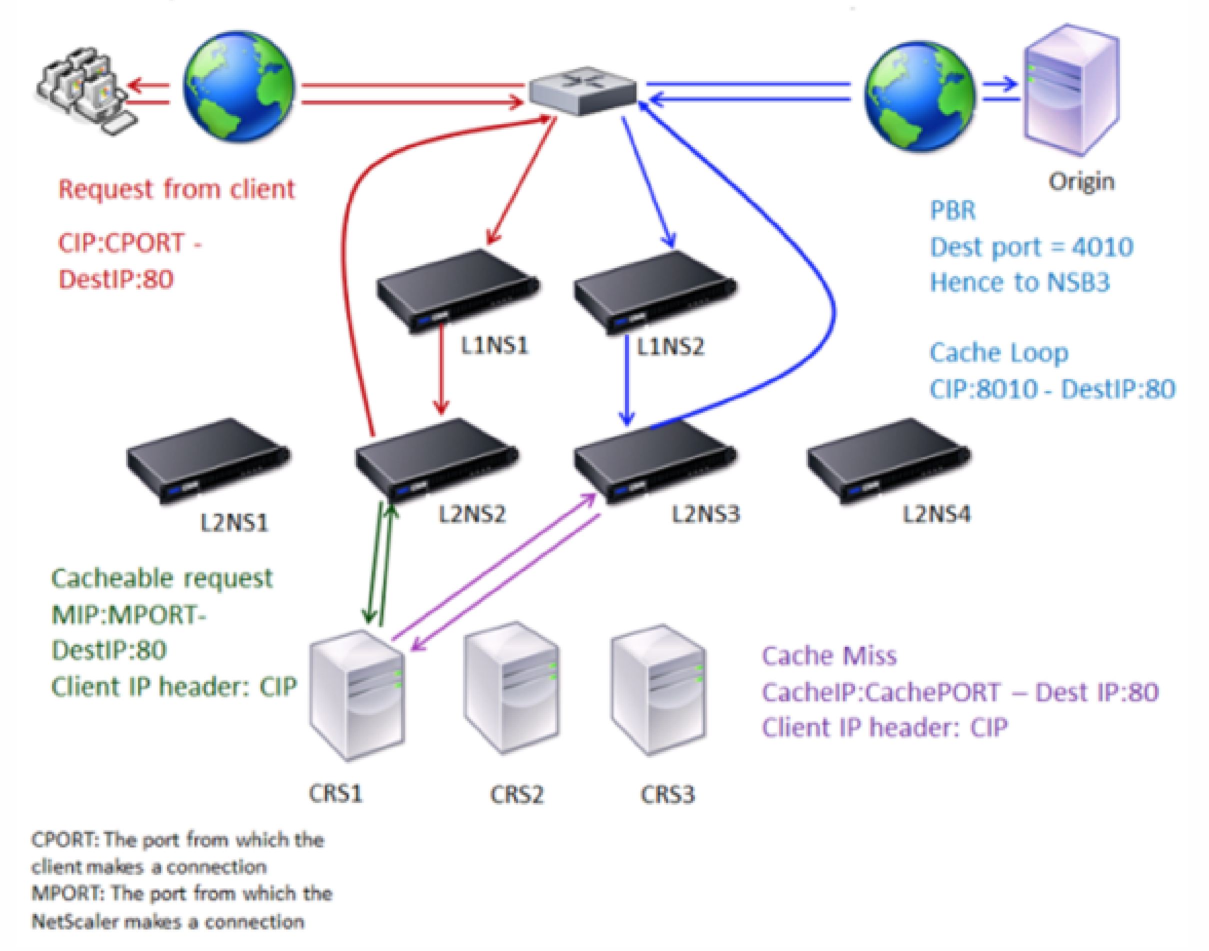
1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. The request is uncacheable (cache bypass). Therefore, L2NS2 sends the request to the origin server through the router.
4. The origin server sends the response to an upper-tier appliance, L1NS2.
5. According to the PBR policies, L1NS2 forwards the traffic to the appropriate appliance in the lower tier, L2NS2.
6. L2NS2 uses the client IP address in the request header to identify the client from which the request came and sends the response directly to the router, avoiding unnecessary load on the appliance in the upper tier.

7. The router forwards the response to Client A.

How N-tier cache redirection works during a cache miss

The following figure shows how cache redirection works when a client request is not cached.

Figure 3. Cache Redirection in Case of a Cache Miss


Request from client
CIP:CPORT -
DestIP:80
Origin
PBR
Dest port = 4010
Hence to NSB3
L1NS1
L1NS2
Cache Loop
CIP:8010 - DestIP:80
L2NS1
L2NS2
L2NS3
L2NS4
Cacheable request
MIP:MPORT-
DestIP:80
Client IP header: CIP
Cache Miss
CacheIP:CachePORT – Dest IP:80
Client IP header: CIP
CRS1
CRS2
CRS3
CPORT: The port from which the
client makes a connection
MPORT: The port from which the
NetScaler makes a connection

Two NetScaler appliances, L1NS1 and L1NS2, are deployed in the upper tier, and four NetScaler appliances, L2NS1, L2NS2, L2NS3, and L2NS4, are deployed in the lower tier. Client A sends a request, which is forwarded by the router. Cache servers CRS1, CRS2, and CRS3 service the cache requests. Origin Server O services the uncached requests.

Traffic flow

1. Client sends a request, and the router forwards it to L1NS1.
2. L1NS1 load balances the request to L2NS2.
3. L2NS2 load balances the request to the cache server CRS1 because the request is cacheable.

4. CRS1 does not have the response (cache miss). CRS1 forwards the request to the origin server through the appliance in the lower tier. L2NS3 intercepts the traffic.
5. L2NS3 takes the client IP from the header and forwards the request to the origin server. The source port included in the packet is the L2NS3 port from which the request is sent to the origin server.
6. The origin server sends the response to an upper-tier appliance, L1NS2.
7. According to the PBR policies, L1NS2 forwards the traffic to the appropriate appliance in the lower tier, L2NS3.
8. L2NS3 forwards the response to the router.
9. The router forwards the response to Client A.

Configure the upper-tier NetScaler appliances

September 21, 2020

Configure each of the upper-tier NetScaler appliances as follows.

Configure an upper-tier appliance for n-tier cache redirection by using the command CLI

At the command prompt, type the following commands:

```
-  add service \<name\>@ \<serviceIP\> \<serviceType\> \<port\>

    Run this command for each service to be added.

-  add lb vserver \<name\>@ ANY \* \<port\> -persistenceType \<
   persistenceMethod\> -lbMethod \<lbMethod\> -m MAC -sessionless
   ENABLED -cltTimeout \<client\_Timeout\_Value\>


-  bind lb vserver \<name\>@ \<serviceName\>

    Run this command for each service to be bound.

-  enable ns mode l3

-  add ns pbr \<name\> \<action\> -srcPort \<sourcePortNumber\> -
   destPort \<startPortNumber-endPortNumber\> -nexthop \<
   serviceIpAddress\> -protocol TCP

```

```
-  apply ns pbrs

     Run this command after adding all the necessary PBRs.
<!--NeedCopy-->
```

Configure an upper-tier appliance for n-tier cache redirection by using the GUI

1. Enable L3 mode:
   a) In the navigation pane, click System, and then click Settings.
   b) In the Settings group, click the Configure modes link.
   c) Select the Layer 3 Mode (IP Forwarding) check box.
   d) Click OK.
2. Configure policy-based routing (PBR):
   a) Navigate to System > Network > PBRs.
   b) In the Policy-Based Routing (PBRs) pane, click Add.
   c) Type a name for the PBR.
   d) Select the action as Allow.
   e) In the Next Hop box, type the IP address of the service, which represents a lower-tier appliance.
   f) Select TCP from the Protocol drop-down list.
   g) Type the source port and the range of the destination port corresponding to the lower-tier appliance being added.
   h) Click Create.
   i) In the details pane, select the PBR and click Apply.
   j) Repeat Step (i) to Step (vii) for each lower-tier appliance.
3. Create a service for each lower-tier appliance:
   a) Navigate to Traffic Management > Load Balancing > Services.
   b) In the details pane, click Add.
   c) Specify the name, protocol, IP address, and port. The protocol should be ANY.
   d) Click Create.
4. Configure a load balancing virtual server:
   a) Navigate to Traffic Management > Load Balancing > Virtual Servers.
   b) In the details pane, click Add.
   c) Specify the name, protocol, IP address, and port. The protocol should be ANY and the IP address should be *.
   d) In the Services tab, select the services that represent the lower-tier NetScaler appliances.
   e) In the Advanced tab, select the Redirection Mode as MAC Based and select the Sessionless check box.
   f) Click Create.

Configure the lower-tier NetScaler appliances

September 21, 2020

Configure each of the lower-tier NetScaler appliances as follows.

Configure a lower-tier appliance for n-tier cache redirection by using the CLI

At the command prompt, type the following commands:

```
-  add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP"  – cachetype transparent

    Repeat for each cache server.

-  add lb vserver <name>@ <serviceType> -m MAC

-  bind lb vserver <name>@ <cacheServiceName>

    Repeat for each cache server.

-  add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ.HEADER("ClientIP")" -originusip ON  – usePortRange ON

-  set ns param-crPortRange <startPortNumber-endPortNumber>
<!--NeedCopy-->
```

Configure a lower-tier appliance for n-tier cache redirection by using the GUI

1. Create a service for each cache server. To create a service:
    a) Navigate to Traffic Management > Load Balancing > Services.
    b) In the details pane, click Add, and specify the name and protocol. Clear the Directly Addressable check box.
    c) In the Advanced tab, select the Override Global check box and the Client IP check box, and then in the Header box, type ClientIP.
    d) In the Cache Type box, select Transparent Cache.
    e) Click Create.
2. Configure a load balancing virtual server:
    a) Navigate to Traffic Management > Load Balancing > Virtual Services.
    b) In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be an asterisk (*).

   c) In the Services tab, select the services that represent the cache servers.
   d) In the Advanced tab, for Redirection Mode, select MAC Based.
   e) Click Create.
3. Configure a cache redirection virtual server:
   a) Navigate to Traffic Management > Load Balancing > Virtual Services.
   b) In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be *.
   c) For Cache Type, select Transparent.
   d) On the Advanced tab, in the Cache Server box, select the new load balancing virtual server and check the Origin USIP and Use Port Range check boxes. In the Source IP Expression box, type HTTP.REQ.HEADER("ClientIP").
   e) Click Create.
4. Assign a source port range for the appliance:
   a) In the navigation pane, click System, and then click Settings.
   b) In the Settings group, click the Change global system settings link.
   c) In the Cache Redirection Port Range group, specify the port range for the appliance by typing a port number for Start Port and a port number for End Port.
   d) Click OK.

Clustering

September 21, 2020

> **Note**
>
> This feature is available with a NetScaler enterprise or platinum edition license.

A NetScaler cluster is a group of nCore appliances working together as a single system image. Each appliance of the cluster is called a node. The cluster can have one appliance or as many as 32 NetScaler nCore hardware or virtual appliances as nodes.

The client traffic is distributed between the nodes to provide high availability, high throughput, and scalability.

To create a cluster, you must add the appliances as cluster nodes, set up communication between the nodes, set up links to the client and server networks, configure the appliances, and configure the distribution of client and server traffic.

Supportability matrix for NetScaler cluster

February 10, 2021

Clustering in NetScaler appliance supports a wide-spread of features in NetScaler appliance configurations.

The following table lists the NetScaler appliance configurations and also provides the status of the features across releases.

> **Important**
>
> The "Node-level" entry in the table indicates that the feature is supported only on individual cluster nodes.

Cluster Configuration	10.5	11.0	11.1
SSL FIPS	No	No	No
SSL Certificate Bundle	No	No	No
Content switching actions; **Note**: The content switching feature is supported from NetScaler 10.1 onwards.	Yes	Yes	Yes
Policy-based logging for content switching policies	Yes	Yes	Yes

Cluster Configuration	10.5	11.0	11.1
Rate limiting	Yes	Yes	Yes
Action analytics	Yes	Yes	Yes
GSLB	Yes	Yes	Yes
RTSP	No	No	Yes
DNSSEC	No	No	No
DNS64	No	No	No
FTP	No	Yes; **Note**: Supported from NetScaler 11.0 Build 62.x onwards.	Yes
TFTP	No	No	No
Connection mirroring	No	No	No
Integrated caching	Node-level	Node-level	Node-Level
Large shared cache	Node-level	Node-level	Node-Level
Application firewall	Node-level	Yes	Yes
HTTP Denial-of-Service Protection (HDOSP)	Node-level	Node-level	Node-Level
Priority queuing (PQ)	Node-level	Node-level	Node-Level

Cluster Configuration	10.5	11.0	11.1
Sure connect (SC)	Node-level	Node-level	Node-Level
AppQoE	Yes	Yes	Yes
Surge protection	Node-level	Node-level	Node-Level
MPTCP	Yes	Yes	Yes
Striped SNIPs	Yes	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.
MSR	Yes	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.
IS-IS (IPv4 and IPv6)	Yes	Yes	Yes
Jumbo Frames	Yes	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes
IP-IP tunneling	Yes	Yes	Yes

Cluster Configuration	10.5	11.0	11.1
Link load balancing	Yes	Yes	Yes
FIS (Failover Interface Set)	Yes	Yes	Yes
Link redundancy (LR)	Yes	Yes	Yes
NAT46	No	No	No
NAT64	No	No	No
RNAT6	No	No	Yes
LSN/CGNAT	No	No	No
IPv6 ReadyLogo	No	No	No
Traffic domains	Yes	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.	Yes; **Note:** Supported in L2 clusters. Not supported in L3 clusters.
Route monitor	No	No	Yes; Only with DR.
GRE tunneling (CB)	No	No	No
Layer 2 mode	Yes	Yes	Yes
Net profiles	Yes	Yes	Yes
HTTPS callout	Yes	Yes	Yes
AAA-TM	Node-level	Node-level	Yes

Cluster Configuration	10.5	11.0	11.1
AppFlow	Node-level	Node-level	Node-Level
Web Insight	No	No	Yes
HDX Insight	No	No	Yes
Vitrual MAC/VRRP	Yes	Yes	Yes
NetScaler Push	No	No	No
Stateful Connection Failover	No	No	No
Graceful Shutdown	No	No	No
DBS AutoScale	No	No	No
DSR using TOS	No	No	No
Finer Startup-RR Control	Node-level	Node-level	Node-Level
XML XSM	No	No	No
DHCP RA	No	No	No
Bridge Group	Yes; **Note:** Supported from NetScaler 10.5 Build 52.1115.e onwards.	Yes	Yes
Network Bridge	No	No	No

Cluster Configuration	10.5	11.0	11.1
Web Interface on NetScaler appliance (WIonNS)	No	Yes (check FAQ); **Note:** Supported from NetScaler 11.0 Build 62.x onwards.	Yes
EdgeSight Monitoring	No	No	Deprecated
Metrics tables - Local	No	No	No
DNS Caching	Node-level	Node-level	NodeLevel
Call Home	Node-level	Node-level	Node Level
NetScaler Gateway ICA Proxy mode	Node-level	Yes	Yes
NetScaler Gateway (SSL VPN / full VPN and clientless VPN)	Node-level	Node-level	Node Level
CloudBridge Connector	No	No	No
Policy Based Routing (PBR/PBR6)	No	No	Yes

Cluster Configuration	10.5	11.0	11.1			
Subscriber awareness	No	No	No			
Dynamic Routing	No	Yes with v4 protocol support	Yes with v6 protocols (ospfv3, `RIPng`, ISIS6, BGP6) support			
SYSLOG-TCP, load balancing of syslog servers, SNIP support, and FQDN support for syslog	No	No	Yes; **Note:** Supported from NetScaler 11.1 Build 54.16 onwards.			
VXLAN	No	No	No	No	No	No

Additionally, the following NetScaler appliance configurations are supported:

Load balancing, load balancing persistency, DNS load balancing, SIP, maxClient, Spillover (connection and dynamic), Spillover based on bandwidth, DataStream, Compression control, Content filtering, TCP buffering, Cache redirection, distributed denial-of-service (DDoS), Client Keep-alive, Basic networking (IPv4 and IPv6), OSPF (IPv4 and IPv6), RIP (IPv4 and IPv6), RIP (IPv4 and IPv6), VLAN, ICMP, Fragmentation, MBF, ACL, Simple ACL, MSR, Path MTU discovery, IP-IP, SNMP, Policies (classic and advanced), Rewrite, Responder, HTTP callout, Web server logging, Audit logging (NSLog and syslog), USIP, Location commands, HTML injection, NITRO API, AppExpert, KRPC.

Prerequisites for cluster nodes

September 21, 2020

NetScaler appliances that are to be added to a cluster must satisfy the following criteria:

- All appliances must have the same software version and build.
- All appliances must be of the same platform type. This means that a cluster must have either all hardware appliances (MPX) or virtual appliances (VPX) or SDX NetScaler appliance instances.

  > **Note**
  >
  > - For a cluster of hardware appliances (MPX), the appliances must be of the same model type.
  > - For the formation of the heterogeneous cluster, all appliances must be of MPX platform type.
  > - For a cluster of virtual appliances (VPX), the appliances must be deployed on the following hypervisors: XenServer, Hyper-V, VMware ESX, and KVM.
  > - Clusters of SDX NetScaler appliance instances are supported in NetScaler 10.1 and later releases. To create a cluster of SDX NetScaler appliance instances, see "Set up a cluster of Citrix NetScaler instances".
  > - Jumbo frames are supported on a NetScaler cluster that is made up of NetScaler appliance SDX instances.
  > - You can create L3 clusters of SDX instances.

- (For releases prior to NetScaler 11.0) All appliances must be on the same network. In NetScaler 11.0 and later releases, appliances can belong to different networks.
- All appliances must have the same licenses. Also, depending on the NetScaler appliance version, there are some additional aspects to address:
  - For releases prior to NetScaler 10.5 Build 52.x:
    - A separate cluster license file is required. This file must be copied to the /nsconfig/license/ directory of the configuration coordinator.
    - Because of the separate cluster license file, the cluster feature is available irrespective of the NetScaler appliance license.
  - For releases after NetScaler 10.5 Build 52.x:
    - No separate cluster license is required.
    - Cluster is licensed with the Enterprise and Platinum licenses. Cluster is not available for Standard license.
- Be initially configured and connected to a common client-side and server-side network.

> **Note**
>
> For a cluster of virtual appliances, that has large configurations, it is recommended to use 6 GB RAM for each node of the cluster.

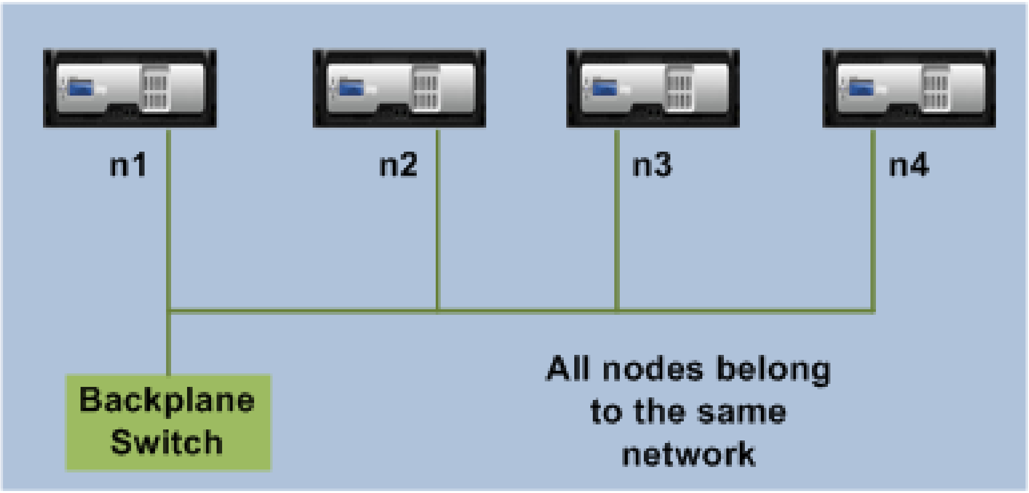
Cluster overview

September 21, 2020

A NetScaler cluster is formed by grouping NetScaler appliances together. Based on the network location of the NetScaler appliances that you intend adding to the cluster, you must be aware of the following cluster setups:

> **Note**
>
> Unless specified otherwise, cluster features and configurations are the same for L2 and L3 clusters.

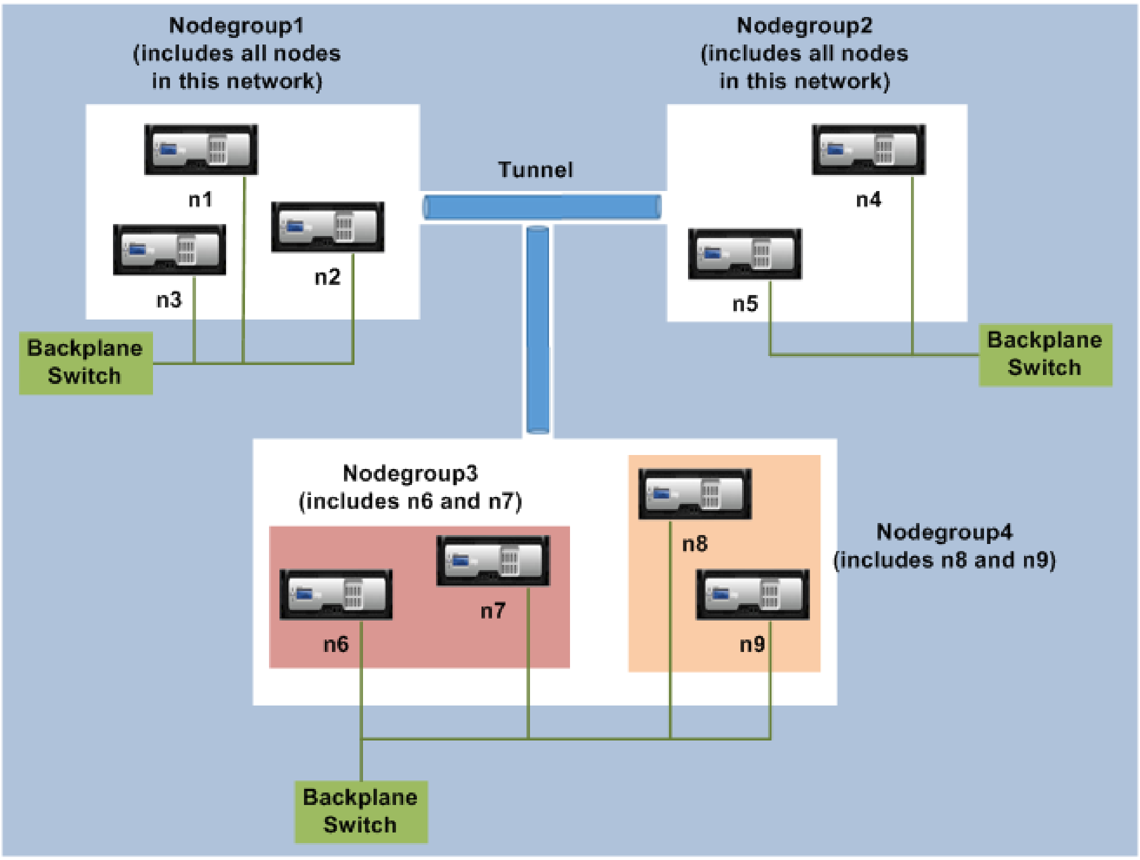
- **L2 cluster:** In this cluster deployment, all cluster nodes belong to the same network.



- **L3 cluster (also referred to as 'cluster in INC mode'):** In this cluster deployment, cluster nodes can belong to different networks. The cluster nodes from a specific network must be grouped into nodegroups that include only nodes from that network. From the following figure, we see that nodes n1, n2, n3 are in the same network and are grouped into Nodegroup1.

  Similarly, the case for nodes n4 and n5, that are grouped in Nodegroup2. In the third network, there are two nodegroups. Nodegroup3 includes n6 and n7 and Nodegroup4 includes n8 and n9.

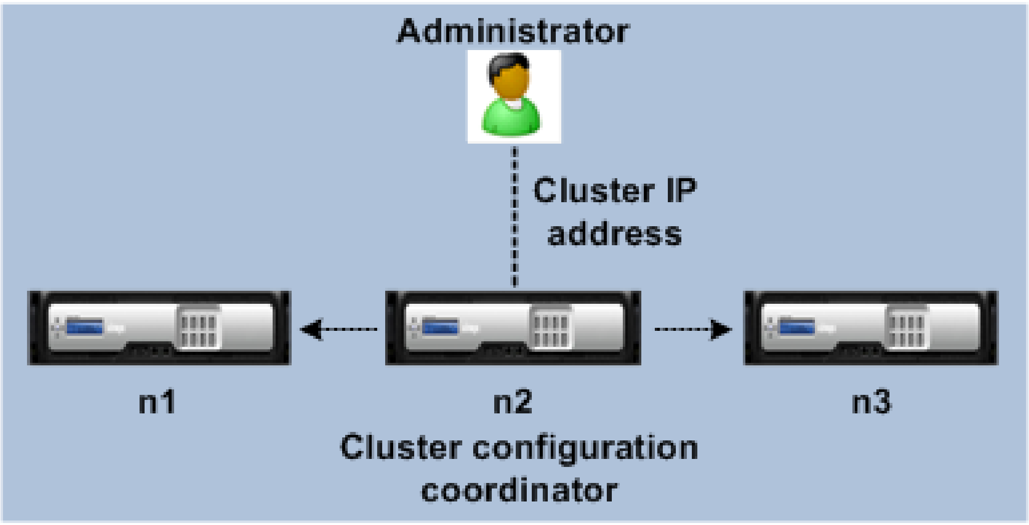
  > **Note**
  >
  > Supported from NetScaler 11.0 onwards.


Nodegroup1
(includes all nodes
in this network)
Nodegroup2
(includes all nodes
in this network)
Tunnel
n1
n2
n3
n4
n5
Backplane Switch
Backplane Switch
Nodegroup3
(includes n6 and n7)
n6
n7
n8
n9
Nodegroup4
(includes n8 and n9)
Backplane Switch

Synchronization across cluster nodes

September 21, 2020

All configurations on a NetScaler cluster are performed on the cluster IP address, which is the management address of the cluster. This cluster IP address is owned by a cluster node that is referred to as the cluster configuration coordinator as shown in the following figure:



The configurations that are available on the configuration coordinator are automatically propagated to the other cluster nodes and therefore all cluster nodes have the same configurations.

- NetScaler appliance allows only a few configurations to be performed on individual cluster nodes through their NSIP address. In these cases, you must ensure configuration consistency manually across all nodes in the cluster. These configurations are not propagated across the other cluster nodes. For more information on operations supported on each cluster nodes, see "Operations Supported on Individual Cluster Nodes".
- The following commands when executed on the cluster IP address are not propagated to other cluster nodes:
  - **shutdown.** Shuts down only the configuration coordinator.
  - **reboot.** Reboots only the configuration coordinator.
  - **rm cluster instance.** Removes the cluster instance from the node that you are executing the command on.
- For a command to propagate to other cluster nodes:
  - The quorum must be configured on cluster instance.
  - The majority of the cluster quorum with (n/2 + 1) of the cluster nodes must be active for the cluster to be operational.
  - A cluster can run with a minimum number of nodes when majority rule (n/2 + 1) is relaxed.

When a node is added to a cluster, the configurations and the files (SSL certificates, licenses, DNS, and so on) that are available on the cluster configuration coordinator are synchronized to the newly added cluster node. When an existing cluster node, that was intentionally disabled or that had failed, is once again added, the cluster compares the configurations available on the node with the configurations available on the configuration coordinator. If there is a mismatch in configurations, the node is synchronized by using one of the following:

- **Full synchronization.** If the difference between configurations exceeds 255 commands, all the configurations of the configuration coordinator are applied to the node that is rejoining the clus-

ter. The node remains operationally unavailable for the duration of the synchronization.
- **Incremental Synchronization.** If the difference between configurations is less than or equal to 255 commands, only the configurations that are not available are applied to the node that is rejoining the cluster. The operational state of the node remains unaffected.

> **Note**
>
> You can also manually synchronize the configurations and files. For more information, see "Synchronizing Cluster Configurations" and "Synchronizing Cluster Files".

Striped, partially striped, and spotted configurations

September 21, 2020

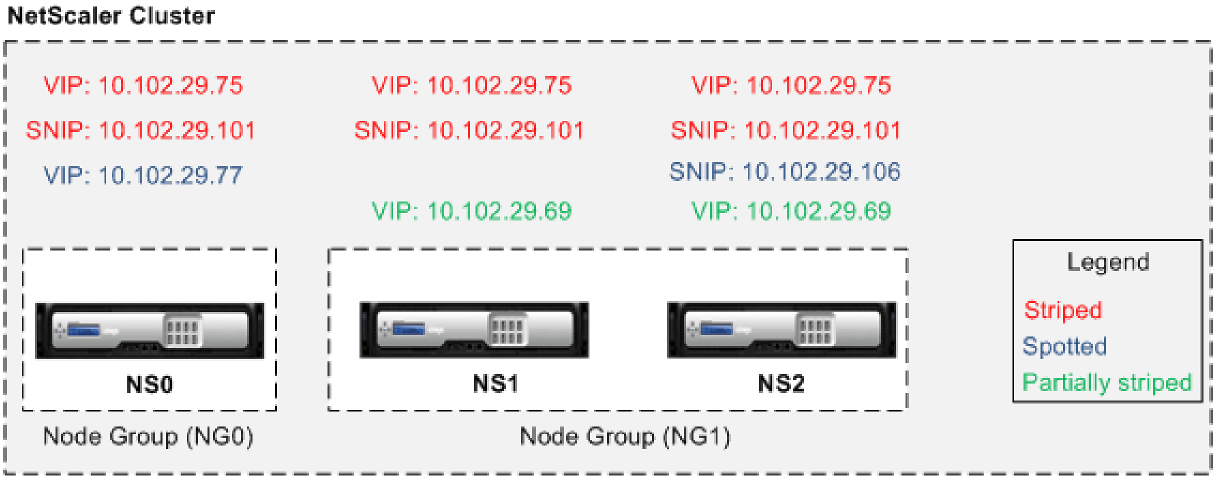
By virtue of command propagation, all nodes in a cluster have the same configurations. While you cannot restrict the nodes on which the configurations are available, you can specify the nodes on which the configurations are active.

For example, you can:

- define a SNIP address to be active on only one node, or
- define a SNIP address to be active on all nodes, or
- define a VIP address to be active on only one node, or
- define a VIP address to be active on all nodes, or
- define a VIP address to be active only on two nodes of a 3-node cluster

Depending on the number of nodes the configurations are active on, cluster configurations are referred to as striped, partially striped, or spotted configurations.

Figure 1. Three-node cluster with striped, partially striped, and spotted configurations


NetScaler Cluster
VIP: 10.102.29.75
SNIP: 10.102.29.101
VIP: 10.102.29.77
VIP: 10.102.29.75
SNIP: 10.102.29.101
VIP: 10.102.29.69
VIP: 10.102.29.75
SNIP: 10.102.29.101
SNIP: 10.102.29.106
VIP: 10.102.29.69
NS0
NS1
NS2
Node Group (NG0)
Node Group (NG1)
Legend
Striped
Spotted
Partially striped

The following table provides more details on the types of configurations:

Configuration Type	Active on	Applicable to	Configurations
Striped configuration	All the cluster nodes	All entries	No specific configuration required to make an entity striped. By default, all entities defined on a cluster IP address are striped on all the cluster nodes.
Partially striped configuration	A subset of cluster nodes	Refer Cluster Node groups.	Bind the entities that you want to be partially striped, to a node group. The configuration is active only on the cluste nodes that belong to the node group.

Configuration Type	Active on	Applicable to	Configurations
Spotted configuration	Single cluster node	SNIP address; SNMP Engine ID; Hostname of cluster nodes; Entities that can be bound to a node group	A spotted configuration can be defined using one of two approaches. **SNIP address** When creating the SNIP address, specify the node on which you want the SNIP address to be active, as the owner node. **Example**, add ns ip 10.102.29.106 255.255.255.0 -type SNIP -owner Node 2 (assuming node NS2 ID is 2). **Note**: You cannot change the ownership of a spotted SNIP address at run time. To change the ownership, you must first delete the SNIP address and add it again by specifying the new owner. **Entities that can be bound to a node group.** By binding the entity to a single-member node group.

> **Note**
>
> When you disable USIP, Citrix recommends you to use spotted SNIP addresses. You can use

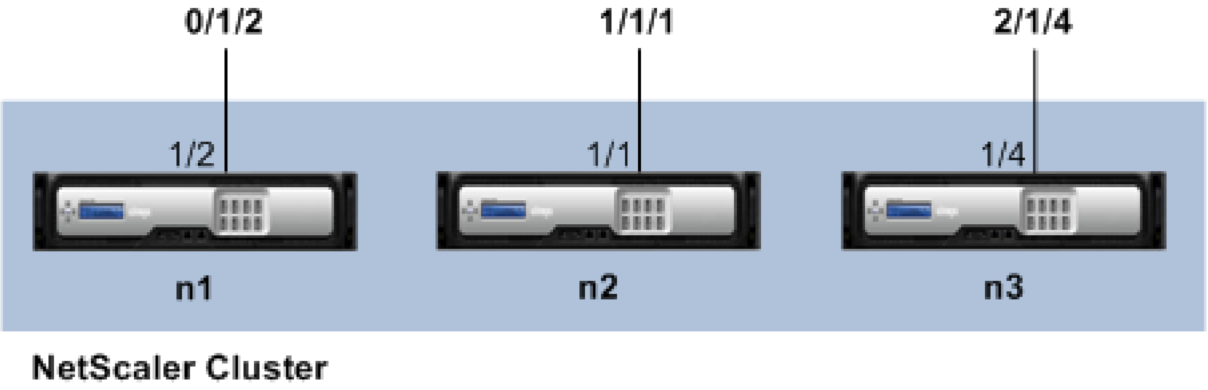
> striped SNIP addresses only if there is a shortage of IP addresses. The use of striped IP addresses can result in ARP flux issues if no spotted IP addresses are present in the same subnet for ARP resolution.
>
> When you enable USIP, Citrix recommends you to use striped SNIP addresses as a gateway for server initiated traffic.

Communication in a cluster setup

September 21, 2020

The interfaces of NetScaler appliances that are added to a cluster, are prefixed with a node ID. This helps identify the cluster node to which the interface belongs. Therefore, the interface identifier c/u, where c is the controller number and u is the unit number, now becomes n/c/u, where n is the node ID. For example, in the following figure, interface 1/2 of node n1 is represented as 0/1/2, interface 1/1 of node n2 is represented as 1/1/1, and interface 1/4 of node n3 is represented as 2/1/4.

Figure 1. Interface naming convention in a cluster



- **Server communication**-
  The cluster communicates with the server through the physical connections between the cluster node and the server-side connecting device. The logical grouping of these physical connections is called the server data plane.
- **Client communication**- The cluster communicates with the client through the physical connections between the cluster node and the client-side connecting device. The logical grouping of these physical connections is called the client data plane.
- **Inter-node communication**- The cluster nodes can also communicate with each other. The manner in which they communicate depends on whether the node exists on the same network or across networks.
  - Cluster nodes within the same network communicate with each other by using the cluster backplane. The backplane is a set of interfaces in which one interface of each node is con-

  nected to a common switch, which is called the cluster backplane switch. The different types of traffic that goes through backplane, which is used by internode communication are:
    - Node to Node Messaging (NNM)
    - Steered traffic
    - Configuration propagation and synchronization
- Each node of the cluster uses a special MAC cluster backplane switch address to communicate with other nodes through the backplane. The cluster special MAC is of the form: (**0x02 0x00 0x6F **), where is the cluster instance ID, is the node number of the NetScaler appliance that are added to a cluster.

The following figures shows the communication interfaces in L2 clusters and L3 clusters.

Figure 2. Cluster communication interfaces - L2 cluster

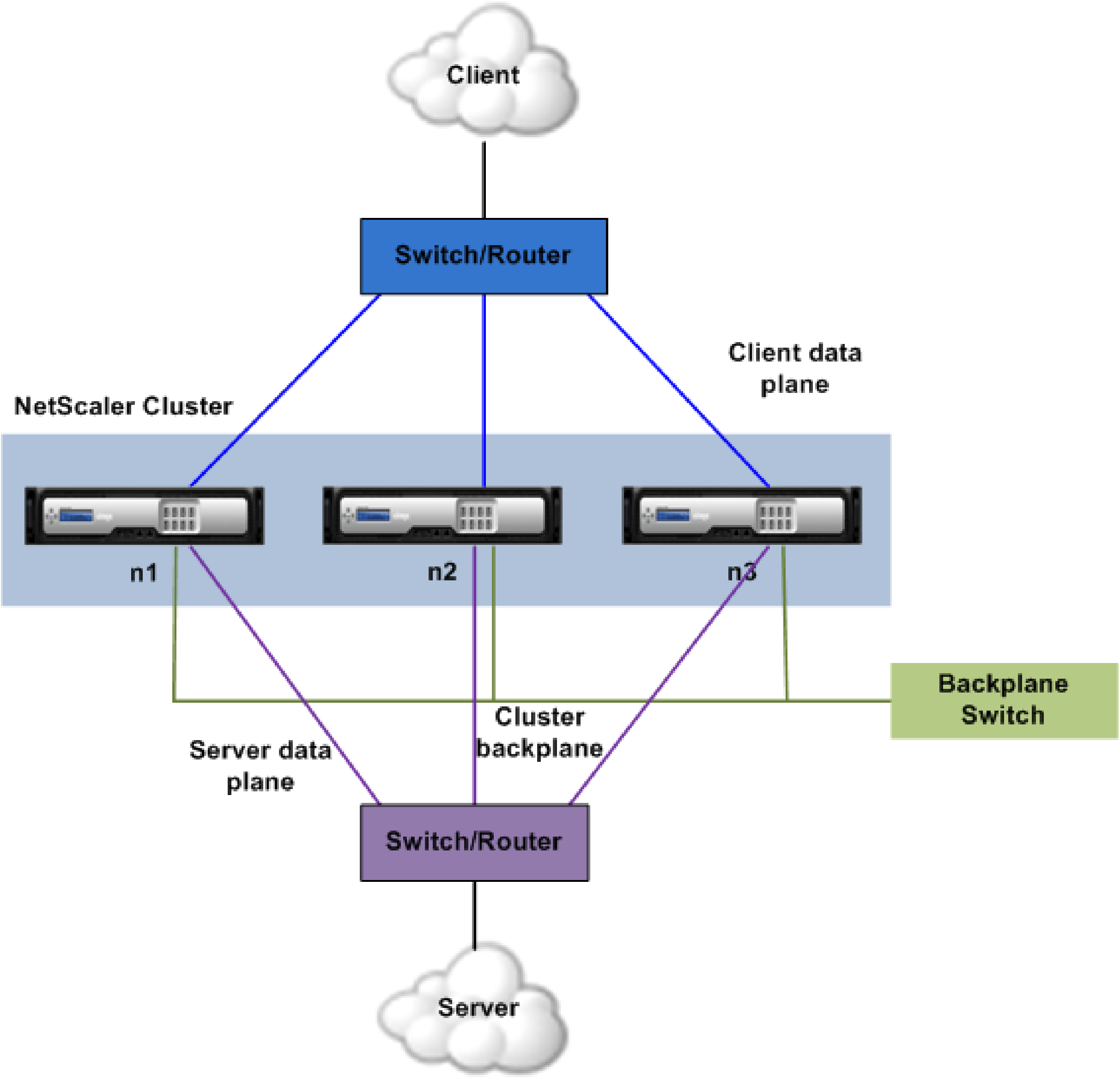
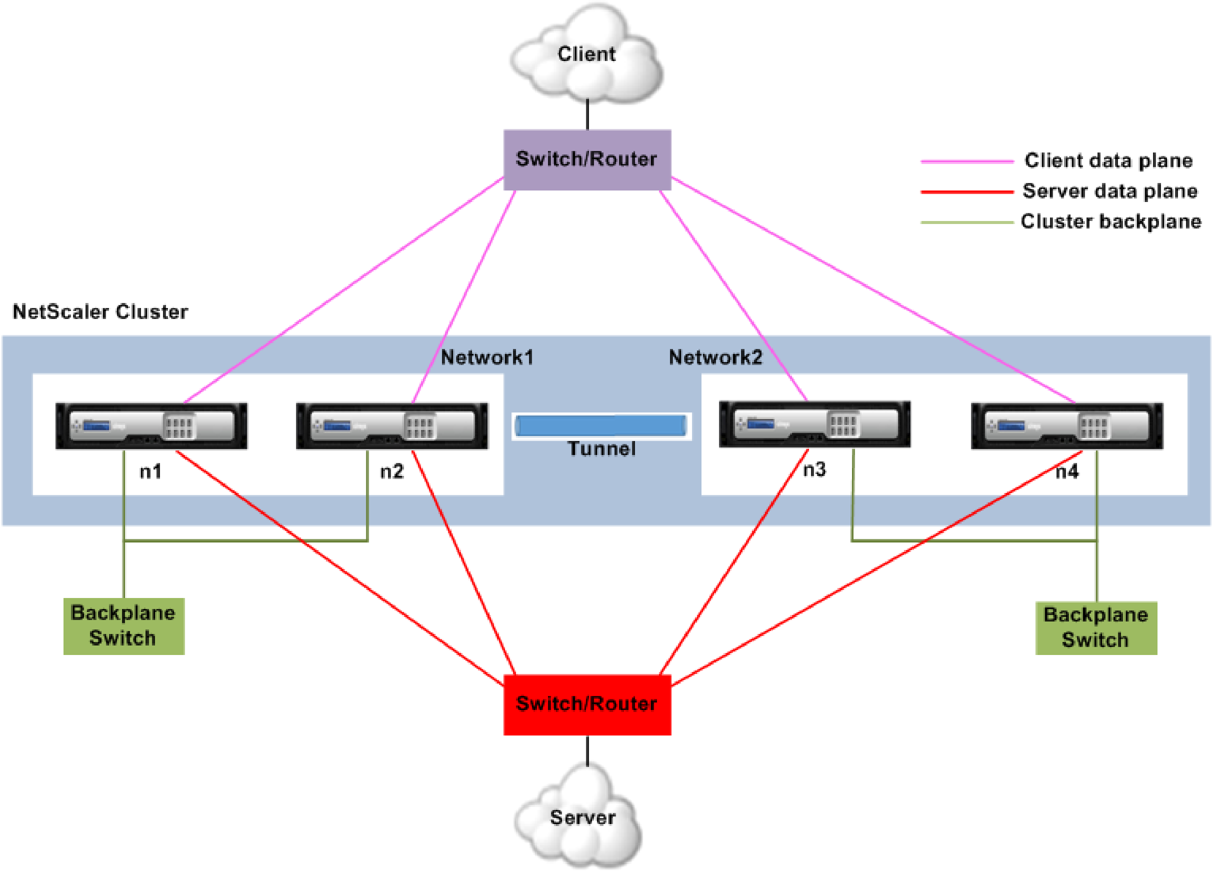

Client
Switch/Router
Client data plane
NetScaler Cluster
n1
n2
n3
Backplane Switch
Cluster backplane
Server data plane
Switch/Router
Server

Figure 3. Cluster communication interfaces - L3 cluster


Client
Switch/Router
Client data plane
Server data plane
Cluster backplane
NetScaler Cluster
Network1
Network2
Tunnel
n1
n2
n3
n4
Backplane Switch
Backplane Switch
Switch/Router
Server

Traffic distribution in a cluster setup

September 21, 2020

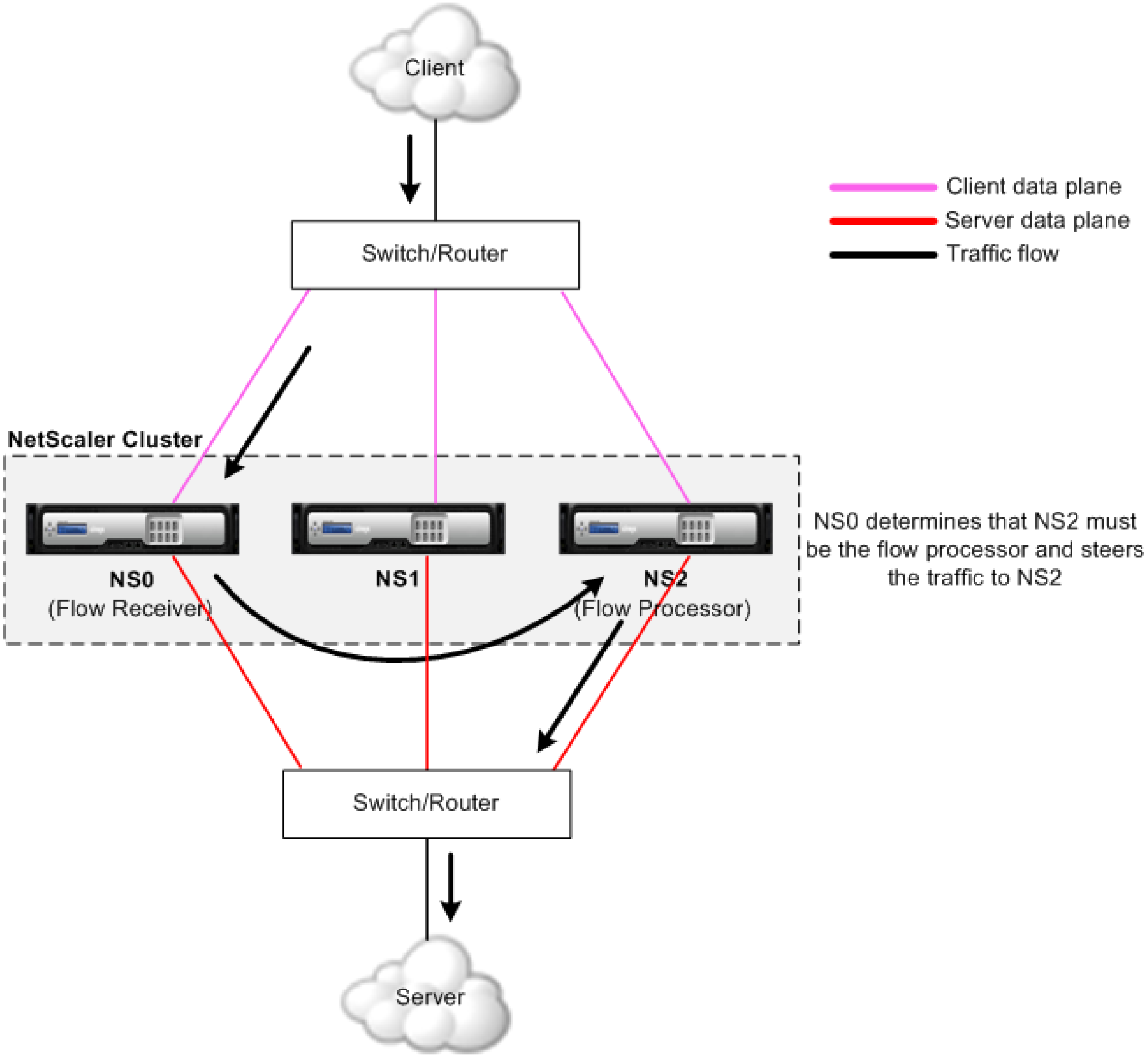
In a cluster setup, external networks view the collection of NetScaler appliances as a single entity. So, the cluster must select a single node that must receive the traffic. The cluster does this selection by using Equal Cost Multiple Path (ECMP) or cluster link aggregation traffic distribution mechanism. The selected node is called the flow receiver.

> **Note**
>
> For an L3 cluster (nodes across different networks), only ECMP traffic distribution can be used.

The flow receiver gets the traffic and then, using internal cluster logic determines the node that must process the traffic. This node is called the flow processor. The flow receiver steers the traffic to the flow processor over the backplane (if the flow receiver and the flow processor are on the same network) or through the tunnel (if the flow receiver and the flow processor are on different networks).

> **Note**
>
> - The flow receiver and flow processor must be nodes capable of serving traffic.
> - From NetScaler 11 onwards, you can disable steering on the cluster backplane. For more information, see "Disabling Steering on the Cluster Backplane".

Figure 1. Traffic distribution in a cluster


Client
Switch/Router
Client data plane
Server data plane
Traffic flow
NetScaler Cluster
NS0
(Flow Receiver)
NS1
NS2
(Flow Processor)
NS0 determines that NS2 must be the flow processor and steers the traffic to NS2
Switch/Router
Server

The above figure shows a client request flowing through the cluster. The client sends a request to a virtual IP (VIP) address. A traffic distribution mechanism configured on the client data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the node that must process the traffic, and steers the request to that node (unless the flow receiver selects itself as the flow processor).

The flow processor establishes a connection with the server. The server processes the request and sends the response to the subnet IP (SNIP) address that sent the request to the server.

- If the SNIP address is a striped or partially striped IP address, the traffic distribution mechanism configured on the server data plane selects one of the cluster nodes as the flow receiver. The flow receiver receives the traffic, determines the flow processor, and steers the request to the flow processor through the cluster backplane.
- If the SNIP address is a spotted IP address, the node that owns the SNIP address receives the response from the server.

In an asymmetric cluster topology (all cluster nodes are not connected to the external switch), you must use linksets either exclusively or combined with ECMP or cluster link aggregation. For more information, see "Using Linksets".

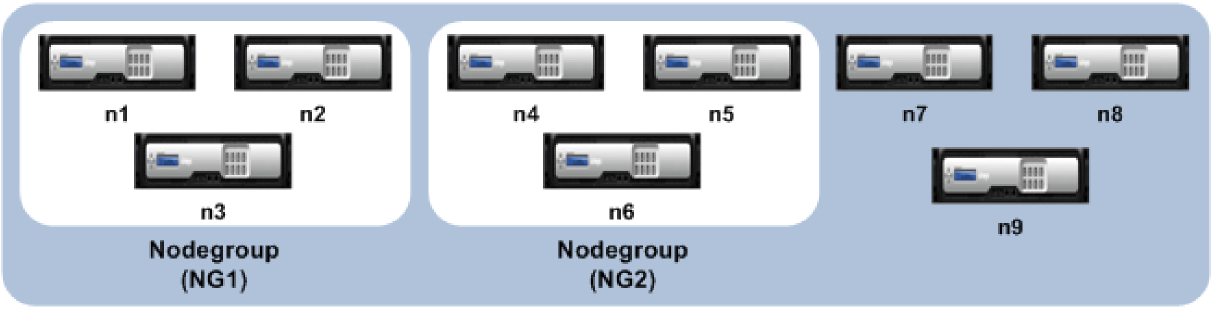
Cluster nodegroups

September 21, 2020

> **Note**
>
> Nodegroups are supported from NetScaler 10.1 onwards.

As the name indicates, a cluster nodegroup is a group of cluster nodes.

Figure 1. NetScaler cluster with nodegroups



The above figure shows a cluster which has nodegroups NG1 and NG2 that include 3 cluster nodes each. The cluster also has 3 nodes that are not part of any nodegroup.

A nodegroup can be configured for the following:

- To define spotted and partially striped configurations. For more information, see "Nodegroups for Spotted and Partially-Striped Configurations".
- To configure redundancy of nodegroups. For more information, see "Configuring Redundancy for Nodegroups".
  Note: Supported from NetScaler 10.5 Build 52.1115.e onwards.

- To define an L3 cluster (also called a cluster in INC mode). In an L3 cluster, cluster nodes can be from different networks. You must group nodes that belong to a network in a single nodegroup. For example, if n1, n2, n3 are in network1 and n4, n5, n6 are in network2, then NG1 must include nodes of network1 and NG2 must include nodes of network2. For setting up an L3 cluster, see "Creating a NetScaler cluster".

> **Note**
>
> - Supported from NetScaler 11 onwards.
> - The above functions of a nodegroup are mutually exclusive. This means that a nodegroup can provide only one of the above mentioned functionality.

Cluster and node states

September 21, 2020

For a cluster to be functional, a majority of the nodes (n/2 + 1) must be operationally active (operational state is ACTIVE). Check table below.

> **Important**
>
> From NetScaler release 10.5, you can configure the cluster to be functional even when the majority criteria is not satisfied. This configuration must be performed when creating a cluster.

For more information on states of a cluster node, refer to
States of a cluster node.

Routing in a cluster

September 21, 2020

Routing in a cluster works in much the same way as routing in a standalone system. A few points to note:

- All routing configurations must be performed from the cluster IP address and the configurations are propagated to the other cluster nodes.
- Routing runs only on spotted SNIP addresses and NSIP addresses.
- Routes are limited to the maximum number of ECMP routes supported by the upstream router.
- Node-specific routing configurations must be performed by using the owner-node argument as follows:

```
!
interface vlan97
!
router ospf
 owner-node 0
  ospf router-id 97.131.0.1
 exit-owner-node
 owner-node 1
  ospf router-id 97.131.0.2
 exit-owner-node
 owner-node 2
  ospf router-id 97.131.0.3
 exit-owner-node
 redistribute kernel
 network 97.0.0.0/8 area 0
!
<!--NeedCopy-->
```

- Retrieve node-specific routing configurations by specifying the node(s) in the owner-node argument as follows:

```
> vtysh
ns# owner-node 0 1
ns(node-0 1)# show cluster state
ns(node-0 1)# exit-owner-node
<!--NeedCopy-->
```

- Clear node-specific routing configurations by specifying the node(s) in the owner-node argument as follows:

```
> vtysh
ns# owner-node 0 1
ns(node-0 1)# clear config
ns(node-0 1)# exit-owner-node
<!--NeedCopy-->
```

- Routing protocol daemons can run and adjacencies can be formed on active and inactive nodes of a cluster.
- Only active nodes advertise host routes to striped VIP addresses. Spotted VIP addresses are advertised by active owner node.
- Active and inactive nodes can learn dynamic routes and install them into the routing table.
- Routes learnt on a node are propagated to other nodes in the cluster only if route propagation

is configured. This is mostly needed in asymmetric topologies where the unconnected nodes may not be able to form adjacencies.

```
ns(config)# ns route-install propagate
<!--NeedCopy-->
```

> **Note***
>
> Make sure that route propagation is not configured in a symmetric cluster topology as it can result in making the node unavailable to the cluster.

Setting up a NetScaler cluster

September 21, 2020

NetScaler appliances that you want to add to the cluster must satisfy the criteria specified in "Prerequisites for Cluster Nodes". Before actually setting up a cluster, you must be aware of cluster basics. For information, see "Cluster Overview".

Forming a cluster requires you to set up inter-node communication, create the cluster (by adding the first NetScaler appliance), and then add the other cluster nodes. Each of these steps is explained with relevant details in subsequent topics.

> **Note**
>
> While there are some differences in setting up an L2 and L3 cluster, there are many similarities too. The subsequent topics explain the setup for both cluster types while highlighting the configurations that are specific to L3 clusters.

Setting up inter-node communication

September 21, 2020

The nodes in a cluster setup communicate with one another using the following inter-node communication mechanisms:

- Nodes that are within the network (same subnet) communicate with each other through the cluster backplane. The backplane must be explicitly set up. See the detailed steps listed below.
- Across networks, steering of packets is done through a GRE tunnel and other node-to-node communication is routed across nodes as required.

> **Important**
>
> - From Release 11.0 all builds, a cluster can include nodes from different networks.
> - From Release 11.1 build 64.11, GRE steering is supported on Fortville NICs in an L3 cluster.
> - In an L3 cluster deployment, packets between NetScaler appliance nodes are exchanged over an unencrypted GRE tunnel that uses the NSIP addresses of the source and destination nodes for routing. When this exchange occurs over the internet, in the absence of an IPsec tunnel, the NSIPs is exposed on the internet, and this can result in security issues. Citrix advises customers to establish their own IPsec solution when using a L3 cluster.

To set up the cluster backplane, do the following for every node

1. Identify the network interface that you want to use for the backplane.
2. Connect an Ethernet or optical cable from the selected network interface to the cluster backplane switch.

For example, to use interface 1/2 as the backplane interface for node 4, connect a cable from the 1/2 interface of node 4 to the backplane switch.

Important points to note when setting up the cluster backplane

- Do not use the appliance's management interface (0/x) as the backplane interface. In a cluster, the interface 0/1/x is read as:

  0 -> node ID 0
  1/x -> NetScaler appliance interface

- Do not use the backplane interfaces for the client or server data planes.

- Configure a link aggregate (LA) channel to optimize the throughput of the cluster backplane.

- In a two-node cluster, where the backplane is connected back-to-back, the cluster is operationally DOWN under any of the following conditions:

  - One of the nodes is rebooted.
  - Backplane interface of one of the nodes is disabled.

  Therefore, Citrix recommends that you dedicate a separate switch for the backplane, so that the other cluster node and traffic are not impacted. You cannot scale out the cluster with a back-to-back link. You might encounter a downtime in the production environment when you scale out the cluster nodes.

- Backplane interfaces of all nodes of a cluster must be connected to the same switch and bound to the same L2 VLAN.

- If you have multiple clusters with the same cluster instance ID, make sure that the backplane interfaces of each cluster are bound to a different VLAN.
- The backplane interface is always monitored, regardless of the HA monitoring settings of that interface.
- The state of MAC spoofing on the different virtualization platforms can affect the steering mechanism on the cluster backplane. Therefore, make sure the appropriate state is configured:
  - XenServer - Disable MAC spoofing
  - Hyper-V - Enable MAC spoofing
  - VMware ESX - Enable MAC spoofing (also make sure “Forged Transmits” is enabled)
- The MTU for the cluster backplane is automatically updated. However, if jumbo frames are configured on the cluster, the MTU of the cluster backplane must be explicitly configured. The value must be set to 78 + X, where X is the maximum MTU of the client and server data planes. For example, if MTU of server data plane is 7500 and of the client data plane is 8922, then the MTU of cluster backplane must be set to 78 + 8922 = 9000. To set this MTU, use the following command:

  ```
  > set interface <backplane_interface> -mtu <value>
  <!--NeedCopy-->
  ```

- The MTU for interfaces of the backplane switch must be specified to be greater than or equal to 1578 bytes, if the cluster has features like MBF, L2 policies, ACLs, routing in CLAG deployments, and vPath.

Creating a NetScaler cluster

September 21, 2020

To create a cluster, start by taking one of the NetScaler appliances that you want to add to the cluster. On this node, you must create the cluster instance and define the cluster IP address. This node is the first cluster node and is called the cluster configuration coordinator. All configurations that are performed on the cluster IP address are stored on this node and then propagated to the other cluster nodes.

The responsibility of configuration coordination in a cluster is not fixed to a specific node. It can change over time depending on the following factors:

- The priority of the node. The node with the highest priority (lowest priority number) is made the configuration coordinator. Therefore, if a node with a priority number lower than that of the existing configuration coordinator is added, the new node takes over as the configuration coordinator.

> **Note**
>
> Node priority can be configured from NetScaler 10.1 onwards.

- If the current configuration coordinator goes down. The node with the next lowest priority number takes over as the configuration coordinator. If the priority is not set or if there are multiple nodes with the lowest priority number, the configuration coordinator is selected from one of the available nodes.

> **Note**
>
> The configurations of the appliance (including SNIP addresses and VLANs) are cleared by implicitly executing the
> clear ns config extended command. However, the default VLAN and NSVLAN are not cleared from the appliance. Therefore, if you want the NSVLAN on the cluster, make sure it is created before the appliance is added to the cluster.
> For an L3 cluster (cluster nodes on different networks), networking configurations are not cleared from the appliance.
>
> **Important**
>
> HA Monitor (HAMON) on a cluster setup is used to monitor the health of an interface on each node. The HAMON parameter should be enabled on each node to monitor the state of the interface. If the operational state of the HAMON enabled interface goes down due to any reason, the respective cluster node is marked as unhealthy (NOT UP) and that node cannot serve traffic.

To create a cluster by using the command line interface

1. Log on to an appliance (for example, appliance with NSIP address 10.102.29.60) that you want to add to the cluster.
2. Add a cluster instance.

   ```
   add cluster instance <clId> -quorumType <NONE | MAJORITY> -inc <ENABLED
    | DISABLED><!--NeedCopy-->
   ```

   > **Note**
   >
   > - The cluster instance ID must be unique within a LAN.
   > - The `-quorumType` parameter must be set to MAJORITY and not NONE in the following scenarios:
   >   - Topologies which do not have redundant links between cluster nodes. These topologies might be prone to network partition due to a single point of failure.
   >   - During any cluster operations such as node addition or removal.
   > - For an L3 cluster, make sure the `-inc` parameter is set to ENABLED. The `-inc` param-

> eter must be disabled for an L2 cluster.

3. [Only for an L3 cluster] Create a nodegroup. In the next step, the newly added cluster node must be associated with this nodegroup.

   > **Note**
   >
   > This nodegroup will include all or a subset of the NetScaler appliances that belong to the same network.

   `add cluster nodegroup <name><!--NeedCopy-->`

4. Add the NetScaler appliance to the cluster.

   "'add cluster node -state -backplane -nodegroup

   ```
   > **Note** For an L3 cluster:
   >
   >-  The nodegroup parameter must be set to the name of the
      nodegroup created above.
   >-  The backplane parameter is mandatory for nodes that are
      associated with a nodegroup that has more than one node, so
      that the nodes within the network can communicate with each
      other.

   Example:

   Adding a node for an L2 cluster (all cluster nodes are in the same
        network).
   ```

   add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1

   ```
   Adding a node for an L3 cluster which includes a single node from
      each network. Here, you do not have to set the backplane.
   ```

   add cluster node 0 10.102.29.60 -state PASSIVE -nodegroup ng1

   ```
   Adding a node for an L3 cluster which includes multiple nodes from
        each network. Here, you have to set the backplane so that
      nodes within a network can communciate with each other.
   ```

   add cluster node 0 10.102.29.60 -state PASSIVE -backplane 0/1/1 -nodegroup ng1
   "'

5. Add the cluster IP address (for example, 10.102.29.61) on this node.

   add ns ip <IPAddress> <netmask> -type clip

   Example

```
add ns ip 10.102.29.61 255.255.255.255 -type clip
<!--NeedCopy-->
```

6. Enable the cluster instance.

   `enable cluster instance <clId><!--NeedCopy-->`

7. Save the configuration.

   `save ns config<!--NeedCopy-->`

8. Warm reboot the appliance.

   `reboot -warm<!--NeedCopy-->`

   Verify the cluster configurations by using the show cluster instance command. Verify that the output of the command displays the NSIP address of the appliance as a node of the cluster.

9. After the node is UP, login to the CLIP and change RPC credentials for both cluster IP address and Node IP address. For more information about changing an RPC node password, see Change an RPC node password.

To create a cluster by using the configuration utility

1. Log on to an appliance (for example, an appliance with NSIP address 10.102.29.60) that you intend to add to the cluster.
2. Navigate to **System > Cluster.**
3. In the details pane, click the **Manage Cluster** link.
4. In the Cluster Configuration dialog box, set the parameters required to create a cluster. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **Create.**
6. In the Configure cluster instance dialog box, make sure that the Enable cluster instance check box is selected.
7. In the Cluster Nodes pane, select the node and click **Open.**
8. In the Configure Cluster Node dialog box, set the State.
9. Click **OK**, and then click **Save.**
10. Warm reboot the appliance.
11. After the node is UP, login to the CLIP and change RPC credentials for both cluster IP address and Node IP address. For more information about changing an RPC node password, see Change an RPC node password.

Adding a node to the cluster

September 21, 2020

You can seamlessly scale the size of a cluster to include a maximum of 32 nodes. When a NetScaler appliance is added to the cluster, the configurations from that appliance are cleared (by internally executing the clear ns config -extended command). The SNIP addresses, MTU settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

The cluster configurations are then synchronized on this node. There can be an intermittent drop in traffic while the synchronization is in progress.

> **Important**
>
> Before you add a NetScaler appliance to a cluster:
>
> - Set up the backplane interface for the node. Check preceding topic.
> - Check if the licenses that are available on the appliance match those availabe on the configuration coordinator. The appliance is added only if the licenses match.
> - If you want the NSVLAN on the cluster, make sure that the NSVLAN is created on the appliance before it is added to the cluster.
> - <Citrix recommends that you add the node as a passive node. Then, after joining the node to the cluster, complete the node specific configuration from the cluster IP address. Run the force cluster sync command if the cluster has only spotted IP addresses, has L3 VLAN binding, or has static routes.
> - When an appliance with a preconfigured link aggregate (LA) channel is added to a cluster, the LA channel continues to exist in the cluster environment. The LA channel is renamed from LA/x to nodeId/LA/x, where LA/x is the LA channel identifier.

To add a node to the cluster by using the command line interface

1. Log on to the cluster IP address, at the command prompt, do the following:

    - Add the appliance (for example, 10.102.29.70) to the cluster.

        > **Note**
        >
        > For an L3 cluster:
        >
        > - The nodegroup parameter must be set to a nodegroup that has nodes of the same network.
        >   >
        >   >

> > - If this node belongs to the same network as the first node that was added, then configure the nodegroup that was used for that node.
> > - If this node belongs to a different network, then create a nodegroup and bind this node to the nodegroup.
> >- The backplane parameter is mandatory for nodes that are associated with a nodegroup that has more than one node, so that the nodes within the network can communicate with each other.

```
add cluster node <nodeId> <IPAddress> -state <state> -
    backplane <interface_name> -nodegroup <name>

Example:

add cluster node 1 10.102.29.70 -state PASSIVE -backplane
    1/1/1
<!--NeedCopy-->
```

- Save the configuration.

```
save ns config
<!--NeedCopy-->
```

2. Log on to the newly added node (for example, 10.102.29.70) and join the node to the cluster.

```
join cluster -clip <ip_addr> -password <password>

Example:

join cluster -clip 10.102.29.61 -password nsroot
<!--NeedCopy-->
```

3. Perform the following configurations:

   - If the node is added to a cluster that has only spotted IPs, the configurations are synchronized before the spotted IP addresses are assigned to that node. In such cases, L3 VLAN bindings can be lost. To avoid this loss, either add a striped IP or add the L3 VLAN bindings.
   - Define the required spotted configurations.
   - Set the MTU for the backplane interface.

4. Save the configuration.

```
save ns config
<!--NeedCopy-->
```

5. Warm reboot the appliance.

```
reboot -warm
<!--NeedCopy-->
```

6. After the node is UP and sync is successful, change RPC credentials for the node from the cluster IP address. For more information about changing an RPC node password, see Change an RPC node password.

```
set rpcNode <node-NSIP> -password <passwd>

Example:

set rpcNode 192.0.2.4 -password mypassword
<!--NeedCopy-->
```

7. Set the cluster node to Active.

```
set cluster node <nodeID> -state active.

Example:

set cluster node 1 -state active
<!--NeedCopy-->
```

To add a node to the cluster by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, click **Add** to add the new node (for example, 10.102.29.70).
4. In the **Create Cluster Node** dialog box, configure the new node. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **Create**. When prompted to perform a warm reboot, click **Yes.**
6. After the node is UP and sync is successful, change RPC credentials for the node from the cluster IP address. For more information about changing an RPC node password, see Change an RPC node password.
7. Navigate to **System > Cluster > Nodes > Edit**.
8. Modify the State to **ACTIVE** and confirm.

To join a previously added node to the cluster by using the configuration utility

If you have used the command line to add a node to the cluster, but have not joined the node to the cluster, you can use the following procedure.

> **Note**
>
> When a node joins the cluster, it takes over its share of traffic from the cluster and hence an existing connection can get terminated.

1. Log on to the node that you want to join to the cluster (for example, 10.102.29.70).
2. Navigate to **System > Cluster.**
3. In the details pane, under Get Started, click the Join Cluster link.
4. In the Join to existing cluster dialog box, set the cluster IP address and the nsroot password of the configuration coordinator. For a description of a parameter, hover the mouse cursor over the corresponding text box.
5. Click **OK.**

Viewing the details of a cluster

September 21, 2020

You can view the details of the cluster instance and the cluster nodes by logging on to the cluster IP address.

To view details of a cluster instance by using the command line interface

Log on to the cluster IP address and, at the command prompt, type:

```
show cluster instance <clId>
```

> **Note**
>
> When executed from the NSIP address of a cluster node that is not the configuration coordinator, this command displays the status of the cluster on this node.

To view details of a cluster node by using the command line interface

Log on to the cluster IP address and, at the command prompt, type:

```
show cluster node <nodeId>
```

To view details of a cluster instance by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**

3. In the details pane, under **Get Started**, click the **Manage Cluster** link to view the details of the cluster.

To view details of a cluster node by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, click the node for which you want to view the details.

Distributing traffic across cluster nodes

September 21, 2020

After you have created the NetScaler cluster and performed the required configurations, you must deploy Equal Cost Multiple Path (ECMP) or cluster Link Aggregation (LA) on the client data plane (for client traffic) or server data plane (for server traffic). These mechanisms distribute external traffic across the cluster nodes.

Using Equal Cost Multiple Path (ECMP)

September 21, 2020

By using the Equal Cost Multiple Path (ECMP) mechanism on a cluster deployment, active cluster nodes advertise the virtual server IP addresses. The cluster node which receives the advertised traffic steers the traffic to the node that must process the traffic. There can be redundant steering in spotted and partially striped virtual servers. Therefore, from NetScaler 11 onwards, spotted and partially striped virtual server IP addresses advertise the owner nodes, which reduce the redundant steering.

You must have detailed knowledge of routing protocols to use ECMP. For more information, see "Configuring Dynamic Routes. For more information on routing in a cluster, see "Routing in a Cluster".

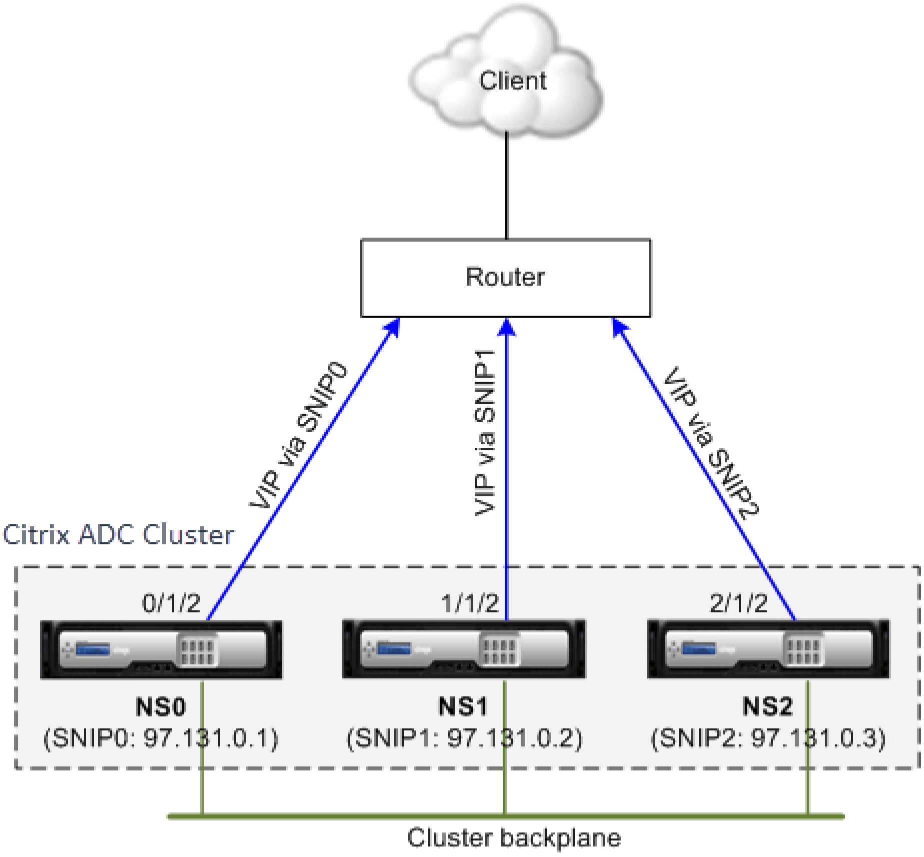
To use ECMP, you must first perform the following:

- Enable the required routing protocol (OSPF, RIP, BGP, or ISIS) on the cluster IP address.
- Bind the interfaces and the spotted IP address (with dynamic routing enabled) to a VLAN.
- Configure the selected routing protocol and redistribute the kernel routes on the ZebOS by using the vtysh shell.

Perform similar configurations on the cluster IP address and on the external connecting device.

> **Note**
>
> - Make sure that the licenses on the cluster support dynamic routing, otherwise ECMP does not work.
> - ECMP is not supported for wildcard virtual servers since RHI needs a VIP address to advertise to a router and wildcard virtual servers do not have associated VIP addresses.

Figure 1. ECMP topology


Client
Router
VIP via SNIP0
VIP via SNIP1
VIP via SNIP2
Citrix ADC Cluster
0/1/2
1/1/2
2/1/2
NS0
(SNIP0: 97.131.0.1)
NS1
(SNIP1: 97.131.0.2)
NS2
(SNIP2: 97.131.0.3)
Cluster backplane

When you use ECMP mechanism for traffic distribution on a cluster deployment, the active cluster nodes advertise the virtual server IP addresses to the upstream router. The ECMP router can reach the VIP address via SNIP0, SNIP1, or SNIP2. The traffic flow in the Figure 1 is described as follows:

1. The client sends a request to the VIP hosted on cluster.
2. The upstream router, based on the learned routes of the VIP, forwards the packet to any one of the nodes. Let's say NS1. The node NS1 is the flow receiver.
3. The flow receiver (NS1) determines the node that must process the traffic, which is called the flow processor. For example, Node NS2 is the flow processor.
4. The flow receiver (NS1) with SNIP1 (97.131.0.2) steers the request to the flow processor (NS2) with SNIP2 (97.131.0.3).
5. The flow processor (NS2) establishes a connection with the server.

6. The server processes the request and sends the response to the SNIP address that sent the request to the server.

Notes:

- Only ACTIVE nodes advertise VIP routes.
- INACTIVE nodes do not advertise VIP routes.
- All ACTIVE nodes advertise striped VIPs.
- Only ACTIVE owner nodes advertise spotted or partially striped VIPs.

To configure ECMP on the cluster by using the command line interface

1. Log on to the cluster IP address.
2. Enable the routing protocol.

   ```
   enable ns feature <feature>
   ```

   Example: To enable the OSPF routing protocol.

   ```
   enable ns feature ospf
   ```

3. Add a VLAN.

   ```
   add vlan <id>
   ```

   Example

   ```
   add vlan 97
   ```

4. Bind the interfaces of the cluster nodes to the VLAN.

   ```
   bind vlan <id> -ifnum <interface_name>
   ```

   Example

   ```
   bind vlan 97 -ifnum 0/1/2 1/1/2 2/1/2
   ```

5. Add a spotted SNIP address for each node and enable dynamic routing on it.

   ```
   add ns ip <SNIP> <netmask> -ownerNode <positive_integer> -
       dynamicRouting ENABLED
   ```

   Example

```
add ns ip 97.131.0.1 255.0.0.0 -ownerNode 0 -dynamicRouting
    ENABLED -type SNIP
add ns ip 97.131.0.2 255.0.0.0 -ownerNode 1 -dynamicRouting
    ENABLED -type SNIP
add ns ip 97.131.0.3 255.0.0.0 -ownerNode 2 -dynamicRouting
    ENABLED -type SNIP
```

6. Bind one of the spotted SNIP addresses to the VLAN. When you bind one spotted SNIP address to a VLAN, all other spotted SNIP addresses defined on the cluster in that subnet are automatically bound to the VLAN.

   ```
   bind vlan <id> -IPAddress <SNIP> <netmask>
   ```

   Example

   ```
   bind vlan 97 -ipAddress 97.131.0.1 255.0.0.0
   ```

   > **Note**
   >
   > You can use NSIP addresses of the cluster nodes instead of adding SNIP addresses. If so, you do not have to perform steps 3–6.

7. Configure the routing protocol on ZebOS using vtysh shell.

   Example:

   To configure an OSPF routing protocol on node IDs 0, 1, and 2.

   ```
   vtysh
   ! interface vlan97 !
    router ospf   owner-node 0
   ospf router-id 97.131.0.1   exit-owner-node
   owner-node 1    ospf router-id 97.131.0.2
   exit-owner-node
   owner-node 2
   ospf router-id 97.131.0.3   exit-owner-node   redistribute kernel
       network 97.0.0.0/8 area 0  !
   ```

   > **Note**
   >
   > For VIP addresses to be advertised, RHI setting is done by using the vserverRHILevel parameter as follows:

   ```
   add ns ip <IPAddress> <netmask> -type VIP -vserverRHILevel <
       vserverRHILevel>
   ```

For OSPF specific RHI settings, there are more settings that can be done as follows:

```
add ns ip <IPAddress> <netmask> -type VIP -ospfLSAType ( TYPE1 |
    TYPE5 ) -ospfArea <positive_integer>
```

Use the add ns ip6 command to perform the preceding commands on IPv6 addresses.

8. Configure ECMP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
//For OSPF (IPv4 addresses) Global config: Configure terminal
    feature ospf    Interface config:  Configure terminal
    interface Vlan10      no shutdown      ip address 97.131.0.5/8
       Configure terminal  router ospf 1  network 97.0.0.0/8 area
    0.0.0.0  ---------------------------------

//For OSPFv3 (IPv6 addresses) Global config:  Configure terminal
    feature ospfv3    Configure terminal  interface Vlan10      no
    shutdown      ipv6 address use-link-local-only      ipv6 router
     ospfv3 1 area 0.0.0.0    Configure terminal  router ospfv3 1
```

Router monitoring cluster nodes in ECMP deployment

In a cluster setup, on an owner node that has a spotted SNIP address configuration, you can now disable the ownerDownResponse option. By default, the option is enabled, allowing the node to respond to an ICMP/ARP/ICMP6/ND6 request coming from the upstream router. You can now disable this option to allow the router to monitor if a cluster node is active or inactive. When the router sends a request, if the option is disabled, it identifies the owner node to be inactive and unavailable for traffic distribution.

To configure ECMP for static routes traffic distribution by using the command line interface

```
add ns ip <ipddress> <netmask> -ownernode <node-id>  - ownerDownResponse
    disable
```

Use Case: ECMP with BGP routing

September 21, 2020

To configure ECMP with BGP routing protocol, perform the following steps:

1. Log on to the cluster IP address.
2. Enable BGP routing protocol.

```
> enable ns feature bgp
<!--NeedCopy-->
```

3. Add VLAN and bind the required interfaces.

```
> add vlan 985
> bind vlan 985 -ifnum 0/0/1 1/0/1
<!--NeedCopy-->
```

4. Add the spotted IP address and bind them to the VLAN.

```
> add ns ip 10.100.26.14 255.255.255.0 -ownerNode 1 -
   dynamicRouting ENABLED
> add ns ip 10.100.26.15 255.255.255.0 -ownerNode 2 -
   dynamicRouting ENABLED
> bind vlan 985 -ipAddress 10.100.26.10 255.255.255.0
<!--NeedCopy-->
```

5. Configure BGP routing protocol on ZebOS using vtysh shell.

```
> vtysh
conf t
router bgp 65535
neighbor 10.100.26.1 remote-as  65535
<!--NeedCopy-->
```

6. Configure BGP on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
router bgp 65535
no synchronization
bgp log-neighbor-changes
neighbor 10.100.26.14 remote-as 65535
neighbor 10.100.26.15 remote-as 65535
  no auto-summary
dont-capability-negotiate
dont-capability-negotiate
no dynamic-capability
<!--NeedCopy-->
```

Using cluster link aggregation

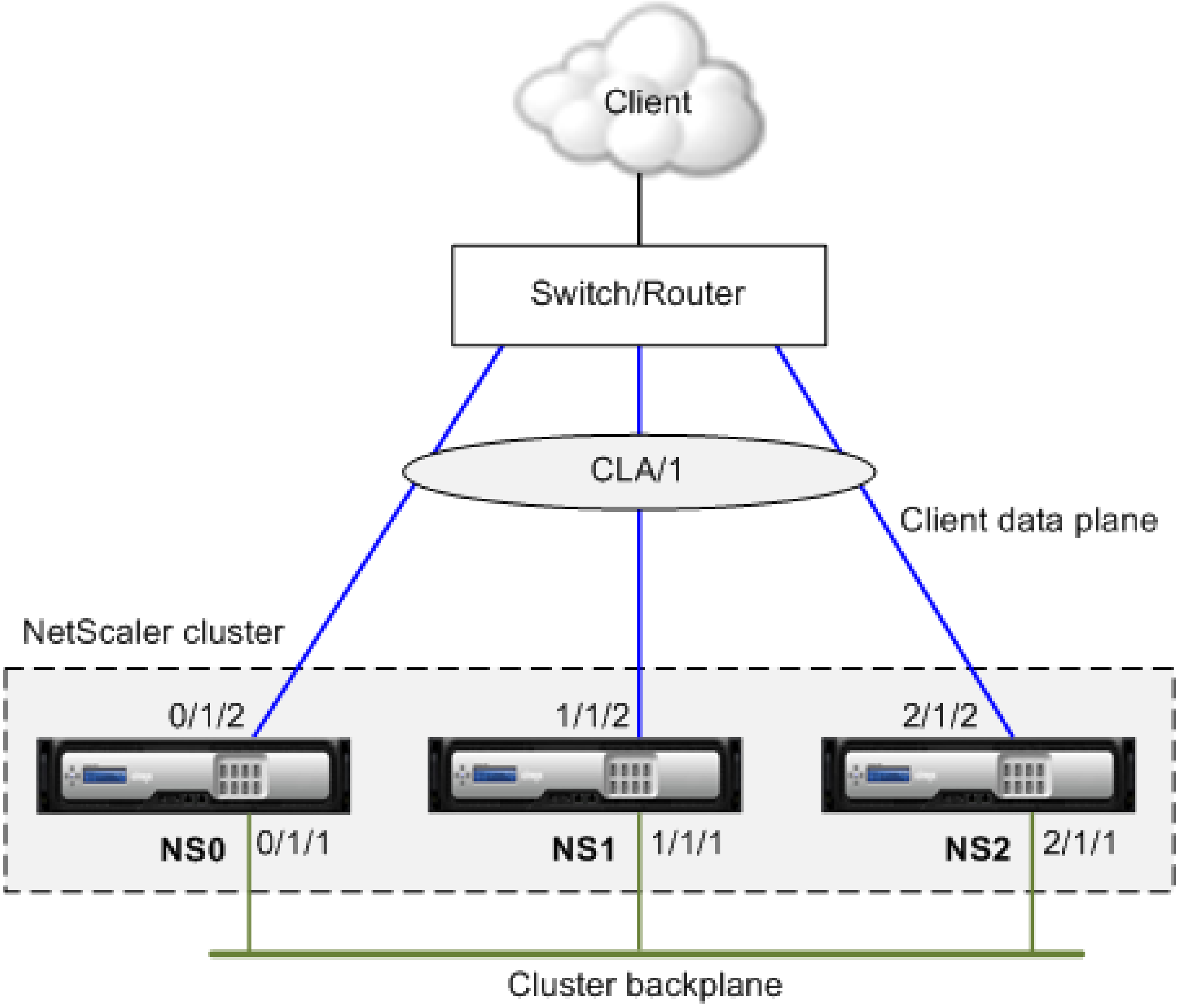
September 21, 2020

Cluster link aggregation, as the name suggests, is a group of interfaces of cluster nodes. It is an extension of NetScaler appliance link aggregation. The only difference is that, while link aggregation requires the interfaces to be from the same device, in cluster link aggregation, the interfaces are from different nodes of the cluster. For more information about link aggregation, see "Configuring Link Aggregation".

> **Important**
>
> - Cluster link aggregation is supported for a cluster of hardware (MPX) appliances.
> - Cluster link aggregation is supported for a cluster of virtual (VPX) appliances that are deployed on ESX and KVM hypervisors, with the following restrictions:
> - Dedicated interfaces need to be used. This means that the interfaces must not be shared with other virtual machines.
> - If the cluster link aggregation member interfaces are manually disabled or if cluster link aggregation itself is manually disabled, then interface power down capability is achieved only by LACP timeout mechanism.
> - Jumbo MTU is not supported on LACP cluster link aggregation.
>
> **Note:** Cluster link aggregation is not supported on VPX appliances that are deployed on XenServer, AWS, and Hyper-V.
>
> - Starting from 12.0 release, cluster link aggregation is supported on NetScaler appliance SDX appliances.
> - The number of interfaces that can be bound to cluster LA is 16 (from each node), and the maximum number of interfaces in cluster LA can be (16 * n), where n is the number of nodes in a cluster. The total number of interfaces in cluster LA depends on the number of interfaces for every port channel on upstream switch.
> - If a NetScaler appliance uses Intel Fortville interfaces, the switchover of a NetScaler cluster node to passive mode might cause a few seconds of outage with CLAG. The issue occurs because LACP is enabled for CLAG to function properly, and the outage time depends on NIC LACP timers.

For example, consider a three-node cluster where all three nodes are connected to the upstream switch. A cluster LA channel (CLA/1) is formed by binding interfaces 0/1/2, 1/1/2, and 2/1/2.

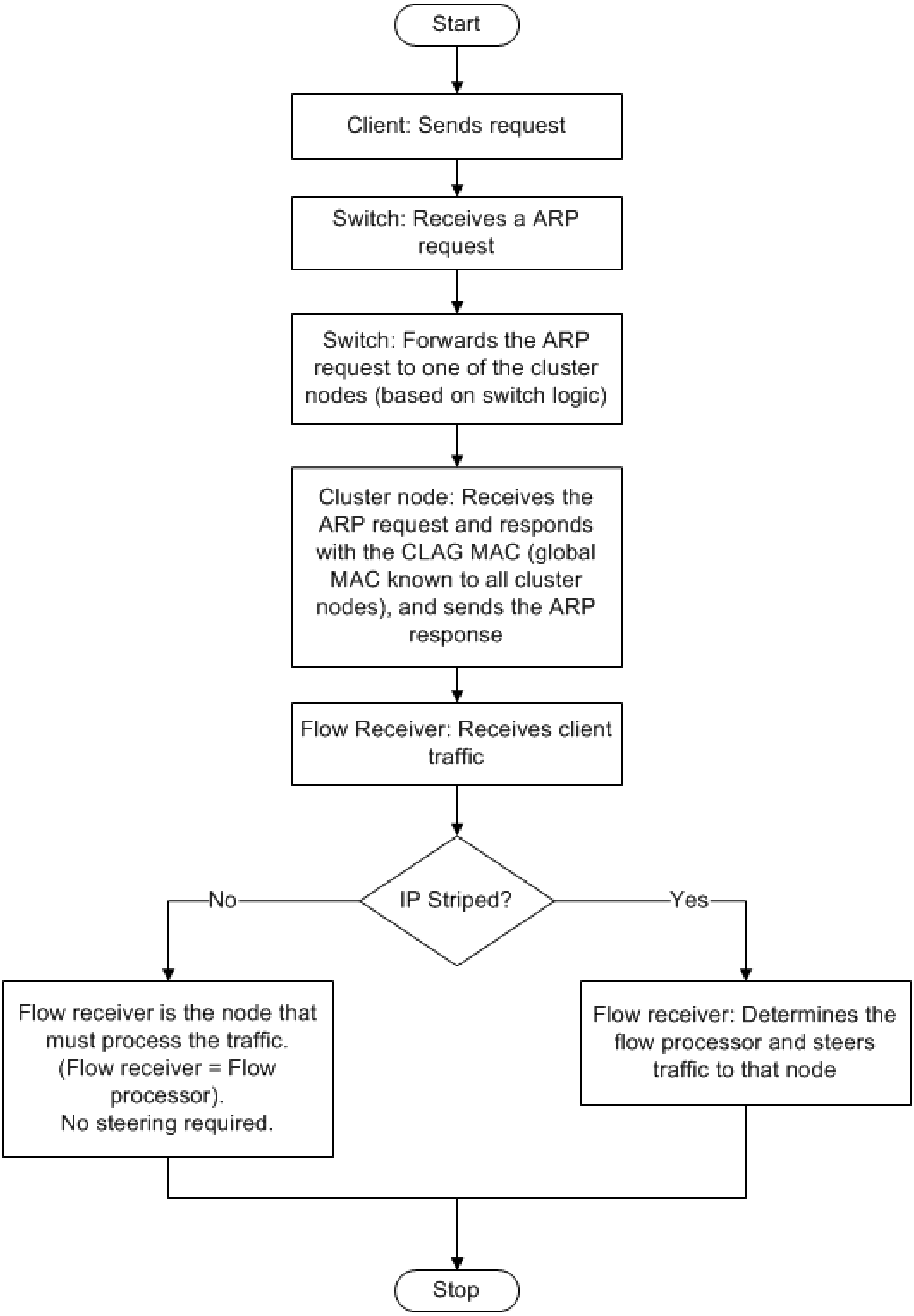
Figure 1. Cluster Link Aggregation topology


Client
Switch/Router
CLA/1
Client data plane
NetScaler cluster
0/1/2
1/1/2
2/1/2
NS0
0/1/1
NS1
1/1/1
NS2
2/1/1
Cluster backplane

A cluster LA channel has the following attributes:

- Each channel has a unique MAC agreed upon by cluster nodes.
- The channel can bind both local and remote nodes' interfaces.
- A maximum of four cluster LA channels are supported in a cluster.
- Backplane interfaces cannot be part of a cluster LA channel.
- When an interface is bound to a cluster LA channel, the channel parameters have precedence over the network interface parameters. A network interface can be bound to one channel only.
- Management access to a cluster node, must not be configured on a cluster LA channel (for example, CLA/1) or its member interfaces. This is because when the node is INACTIVE, the corresponding cluster LA interface is marked as power down and therefore looses management access.

Figure 2. Traffic distribution flow using cluster LA


Start
Client: Sends request
Switch: Receives a ARP request
Switch: Forwards the ARP request to one of the cluster nodes (based on switch logic)
Cluster node: Receives the ARP request and responds with the CLAG MAC (global MAC known to all cluster nodes), and sends the ARP response
Flow Receiver: Receives client traffic
IP Striped?
No
Yes
Flow receiver is the node that must process the traffic. (Flow receiver = Flow processor). No steering required.
Flow receiver: Determines the flow processor and steers traffic to that node
Stop

Static cluster link aggregation

September 21, 2020

You must configure a static cluster LA channel on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

To configure a static cluster LA channel by using the command line interface

1. Log on to the cluster IP address.

   > **Note**
   >
   > Make sure that you configure the cluster LA channel on the cluster IP address before configuring link aggregation on the external switch. Otherwise, the switch will forward traffic to the cluster even though the cluster LA channel is not configured. This can lead to loss of traffic.

2. Create a cluster LA channel.

   “‘add channel -speed

   ```
   **Example**
   ```

   > add channel CLA/1 -speed 1000
   > “‘

   > **Note**
   >
   > You must not specify the speed as AUTO. Rather, you must explicitly specify the speed as 10, 100, 1000, or 10000. Only interfaces that have the speed matching the <speed> attribute in the cluster LA channel are added to the active distribution list.

3. Bind the required interfaces to the cluster LA channel. Make sure that the interfaces are not used for the cluster backplane.

   “‘bind channel

   ```
   **Example**
   ```

   > bind channel CLA/1 0/1/2 1/1/2 2/1/2
   > “‘

4. Verify the configurations.

   “‘show channel

```
**Example**
```

> show channel CLA/1
> “‘

> **Note**
>
> You can bind the cluster LA channel to a VLAN by using the bind vlan command. The interfaces of the channel are automatically bound to the VLAN.

5. Configure static LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.

```

Global config:
Configure terminal

Interface level config:

interface Ethernet2/47
    switchport
    switchport access vlan 10
    channel-group 7 mode on
    no shutdown

interface Ethernet2/48
    switchport
    switchport access vlan 10
    channel-group 7 mode on
    no shutdown
<!--NeedCopy-->
```

Dynamic cluster link aggregation

September 21, 2020

Dynamic cluster LA channel uses Link Aggregation Control Protocol (LACP).

You must perform similar configurations on the cluster IP address and on the external connecting device. If possible, configure the upstream switch to distribute traffic based on IP address or port instead of MAC address.

Points to remember:

- Enable LACP (by specifying the LACP mode as either ACTIVE or PASSIVE).

  > **Note**
  >
  > Make sure the LACP mode is not set as PASSIVE on both the NetScaler cluster and the external connecting device.

- Specify the same LACP key on each interface that you want to be the part of the channel. For creating a cluster LA channel, the LACP key can have a value from 5 through 8. For example, if you set the LACP key on interfaces 0/1/2, 1/1/2, and 2/1/2 to 5, CLA/1 is created. The interfaces 0/1/2, 1/1/2, and 2/1/2 are automatically bound to CLA/1. Similarly, if you set the LACP key to 6, CLA/2 channel is created.
- Specify the LAG type as Cluster.

To configure a dynamic cluster LA channel by using the command line interface

On the cluster IP address, for each interface that you want to add to the cluster LA channel, type:

“‘set interface -lacpMode -lacpKey -lagType CLUSTER

```
**Example:** To configure a cluster LA channel CLA/1 of 3 interfaces.
```

> set interface 0/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 1/1/2 -lacpMode active -lacpKey 5 -lagType Cluster
> set interface 2/1/2 -lacpMode active -lacpKey 5 -lagType Cluster

```
> **Note**
>
> Optionally, you can enable
[Link Redundancy in a Cluster with LACP](/en-us/netscaler/11-1/clustering/cluster-traffic-distribution/cluster-link-aggregation/cluster-link-redundancy.html).

Similarly, configure dynamic LA on the external switch. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1). Similar configurations must be performed on other switches.
```

Global config:
Configure terminal
feature lacp
Interface level config:

```
interface Ethernet2/47
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown

interface Ethernet2/48
switchport
switchport access vlan 10
channel-group 7 mode active
no shutdown
```

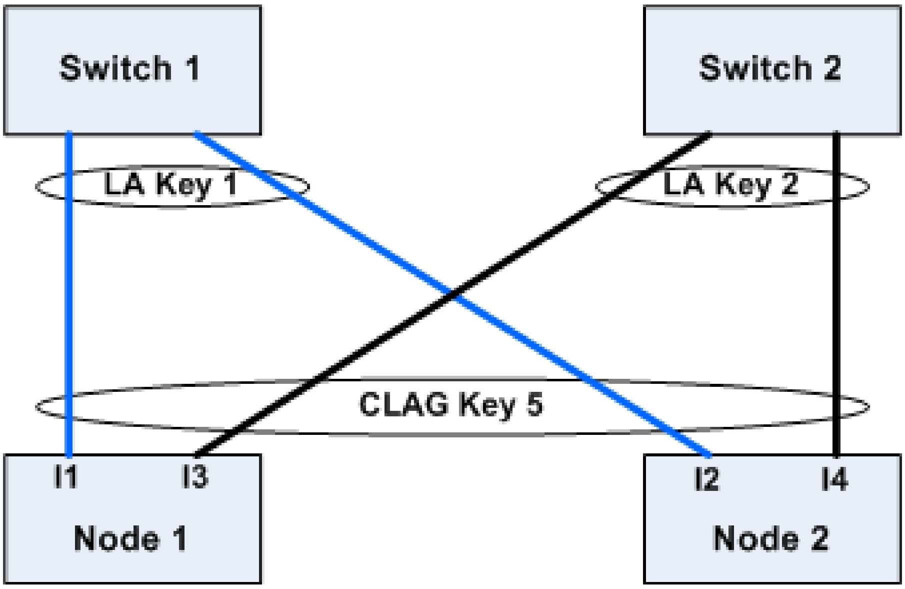
“‘

Link redundancy in a cluster with LACP

September 21, 2020

A NetScaler cluster provides link redundancy for LACP to ensure that all nodes have the same partner key.

To understand the need for link redundancy, let us consider the example of the following cluster setup along with the accompanying cases (with attention to case 3):



In this setup, interfaces I1, I2, I3 and I4 are bound to LACP channel with KEY 5. On the partner side, I1 and I2 are connected to Switch 1 to form a single LA channel with KEY 1. Similarly, I3 and I4 are connected to Switch 2 to form a single LA channel with KEY 2.

Now let us consider the following cases to understand the need for link redundancy:

- **Case 1: Switch 1 is up and Switch 2 is down**

  In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key2 and would start receiving LACPDUs from Key1. On both the nodes, cluster LA is connected to KEY 1 and I1 and I2 will be UP and channel on both the nodes would be UP.

- **Case 2: Switch1 goes down and Switch2 becomes UP**

  In this case, cluster LA on both the nodes would stop receiving LACPDUs from Key1 and would start receiving LACPDUs from Key2. On both the nodes, cluster LA is connected to Key2 and I3 and I4 will be UP and channel on both the nodes would be UP.

- **Case 3: Both Switch1 and Switch2 are UP**

  In this case, it is possible that cluster LA on node1 chooses Key1 as its partner and cluster LA on node2 chooses Key2 as its partner. This means that I1 on node1 and I4 on node2 are receiving traffic which is undesirable. This can happen because the LACP state machine is node-level and chooses its partners on first-come first-serve basis.
  To solve these concerns, link redundancy of dynamic cluster LA is supported. To configure link redundancy on a channel or interface, you must enable it
  and optionally specify the threshold throughput as follows:
  set channel CLA/1 -linkRedundancy ON -lrMinThroughput <positive_integer>
  The throughput of the partner channels is checked against the configured threshold throughput. The partner channel that satisfies the threshold throughput is selected in first-in-first-out (FIFO) manner. If none of the partner channel meets the threshold, or if threshold throughput is not configured, the partner channel with the maximum number of links is selected.

  > **Note**
  >
  > The threshold throughput can be configured from NetScaler 11 onwards.

Managing the NetScaler cluster

September 21, 2020

After you have created a cluster and configured the required traffic distribution mechanism, the cluster is able to serve traffic. During the lifetime of the cluster, you can perform cluster tasks such as configuring nodegroups, disabling nodes of a cluster, discovering NetScaler appliances, viewing statistics, synchronizing cluster configurations, cluster files, and the time across the nodes, and upgrading or downgrading the software of cluster nodes.

Configuring linksets

September 21, 2020

Linksets must be used when some cluster nodes are not physically connected to the external network. In such a cluster topology, the unconnected cluster nodes use the interfaces specified in the linkset to communicate with the external network through the cluster backplane. Linksets are typically used in scenarios when the connecting devices have insufficient ports to connect the cluster nodes.

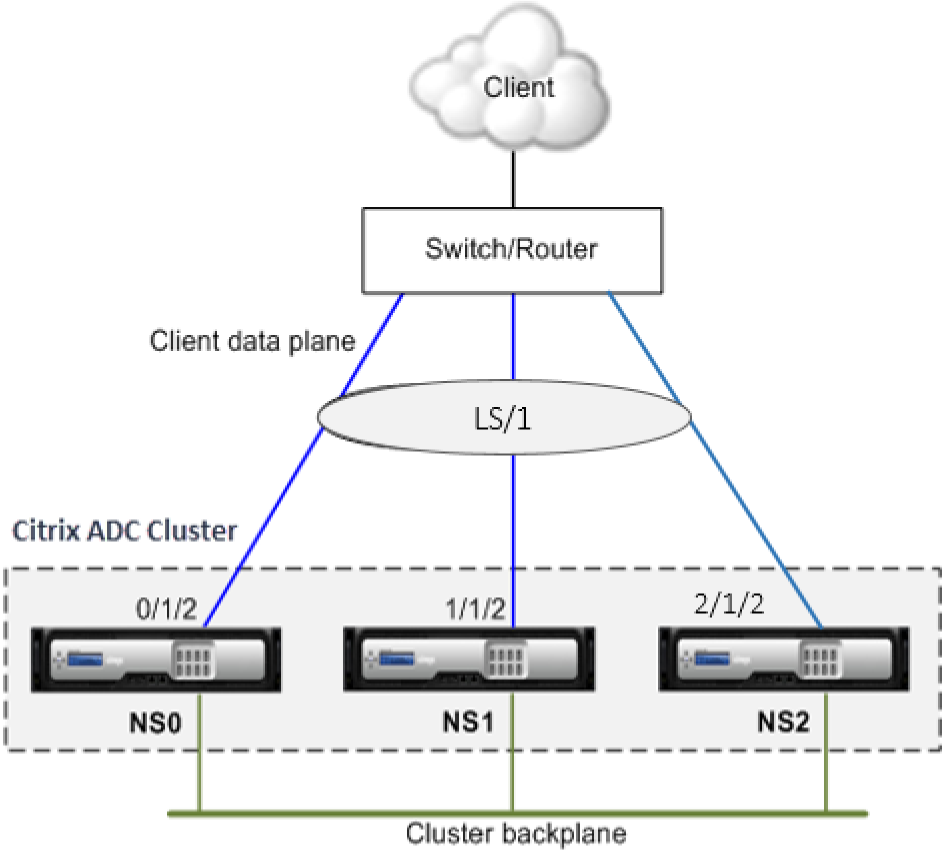
> **Note**
>
> Linksets are a mandatory configuration in the following scenarios:

- For deployments that require MAC-Based Forwarding (MBF).
- To improve manageability of ACL and L2 policies involving interfaces. You must define a linkset of the interfaces and add ACL and L2 policies based on linksets.

Linksets must be configured only through the cluster IP address.

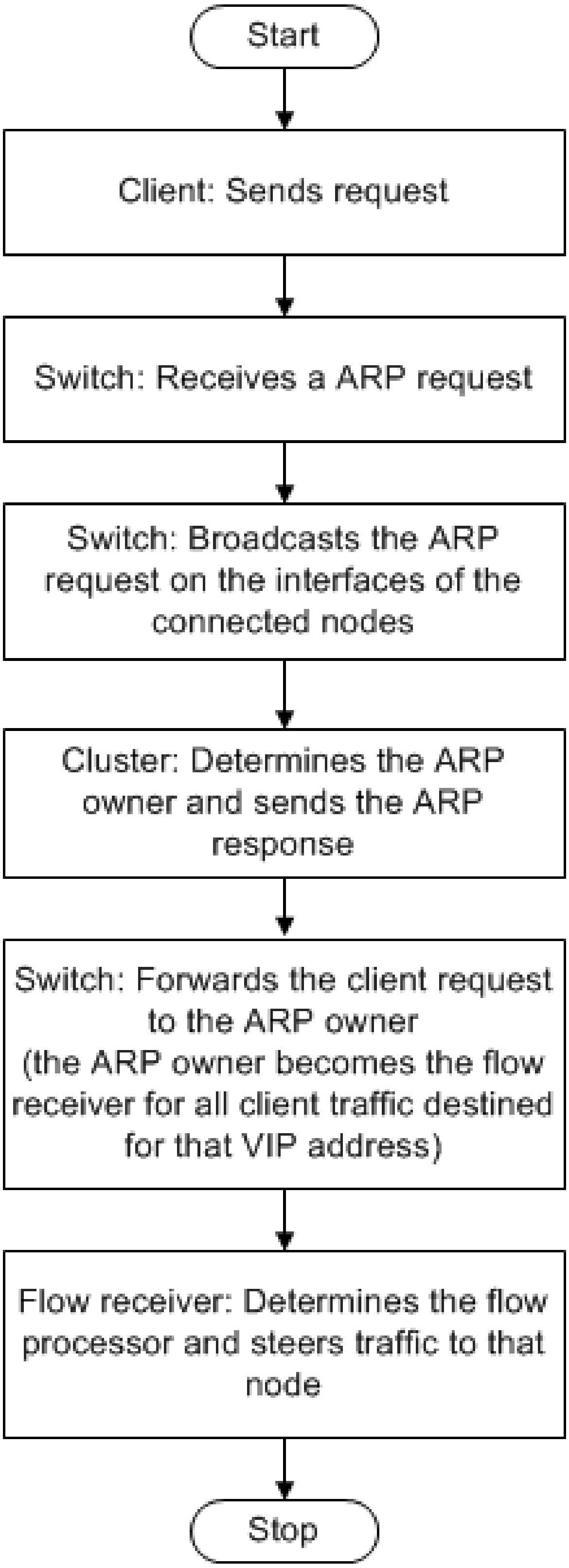
For example, consider a three node cluster where the upstream switch has only two ports available. Using linksets, you can connect two nodes to the switch and leave the third node unconnected. In the following figure, a linkset (LS/1) is formed by binding the interfaces 0/1/2 and 1/1/2. NS2 is the unconnected node of the cluster.

Figure 1. Linksets topology


Client
Switch/Router
Client data plane
LS/1
Citrix ADC Cluster
0/1/2
1/1/2
2/1/2
NS0
NS1
NS2
Cluster backplane

The linkset informs NS2 that it can use interfaces 0/1/2 and 1/1/2 to communicate with the network devices. All traffic to and from NS2 is now routed through interfaces 0/1/2 or 1/1/2.

Figure 2. Traffic distribution flow using linksets


Start
Client: Sends request
Switch: Receives a ARP request
Switch: Broadcasts the ARP request on the interfaces of the connected nodes
Cluster: Determines the ARP owner and sends the ARP response
Switch: Forwards the client request to the ARP owner (the ARP owner becomes the flow receiver for all client traffic destined for that VIP address)
Flow receiver: Determines the flow processor and steers traffic to that node
Stop

To configure a linkset by using the command line interface

1. Log on to the cluster IP address.
2. Create a linkset.

   “‘add linkset

   ```
   **Example**
   ```

   add linkset LS/1
   “‘

3. Bind the required interfaces to the linkset. Make sure the interfaces are not used for the cluster backplane.

   “‘bind linkset -ifnum ...

   ```
   **Example**
   ```

   > bind linkset LS/1 -ifnum 0/1/2 1/1/2
   > “‘

4. Verify the linkset configurations.

   “‘show linkset

   ```
   **Example**
   ```

   > show linkset LS/1
   > “‘

   > **Note**
   >
   > You can bind the linkset to a VLAN by using the bind vlan command. The interfaces of the linkset are automatically bound to the VLAN.

To configure a linkset by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Network > Linksets**.
3. In the details pane, click **Add**.
4. In the **Create Linkset** dialog box:
   - Specify the name of the linkset by setting the Linkset parameter.
   - Specify the Interfaces to be added to the linkset and click Add. Repeat this step for each interface you want to add to the linkset.
5. Click **Create**, and then click **Close**.

Nodegroups for spotted and partially-striped configurations

September 21, 2020

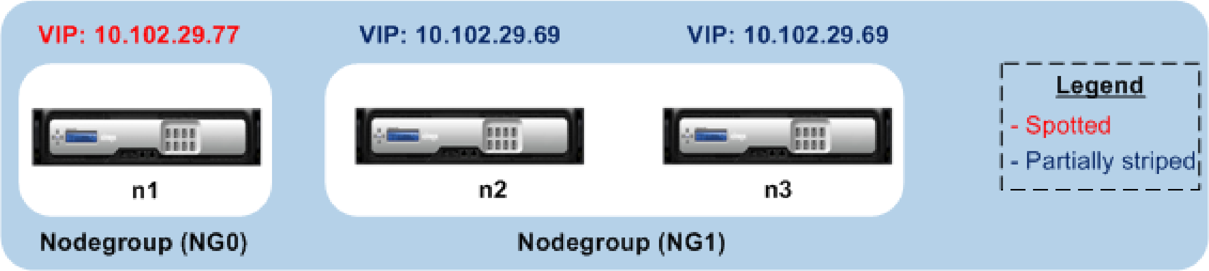
By virtue of the default cluster behavior, all configurations performed on the cluster IP address are available on all nodes of the cluster. However, there might be cases where you need some configurations to be available only on specific cluster nodes.

You can achieve this requirement by defining a nodegroup that includes the specific cluster nodes, and then binding the configuration to that nodegroup. This ensures that the configuration is active only on those cluster nodes. These configurations are called partially-striped or spotted (if active only one a single node). For more information, see Striped, Partially Striped, and Spotted Configurations.

For example, consider a cluster with three nodes. You create a nodegroup NG0 that includes node n1 and another nodegroup NG1 that includes n2 and n3. Bind load balancing virtual servers .77 to NG0 and load balancing virtual server .69 to NG1.

This means that virtual server .77 will be active only on n1 and consequently only n1 will receive traffic that is directed to .77. Similarly, virtual server .69 will be active only on nodes n2 and n3 and consequently only n2 and n3 will receive traffic that is directed to .69.

Figure 1. NetScaler cluster with nodegroups configured for spotted and partial-striped configurations



The entities or configurations that you can bind to a nodegroup are:

- Load balancing, content switching, cache redirection, authentication, authorization, and auditing virtual servers.

  > **Note**
  >
  > FTP load balancing virtual servers cannot be bound to nodegroups.

- VPN virtual server (Supported from NetScaler 10.5 Build 50.10 onwards).
- Global Server Load Balancing (GSLB) sites and other GSLB entities (Supported from NetScaler 10.5 Build 52.11 onwards).
- Limit identifiers and stream identifiers.

Behavior of nodegroups

September 21, 2020

Due to the interoperability of nodegroups with different NetScaler features and entities, there are some behavioral aspects to be noted. Nodes in a nodegroup can also be backed up. Read on for more information.

General behavior of a cluster nodegroup

- A nodegroup that has entities bound to it cannot be removed.
- A cluster node that belongs to a nodegroup with entities bound to it, cannot be removed.
- A cluster instance that has nodegroups with entities bound to it, cannot be removed.
- You cannot add an entity that has a dependency on another entity that is not part of the nodegroup. If you need to do so, first remove the dependency. Then, add both the entities to the nodegroup and reassociate the entities.

  Examples:

  - Assume you have a virtual server, VS1, whose backup is virtual server VS2. To add VS1 to a nodegroup, first make sure that VS2 is removed as the backup server of VS1. Then, bind each server individually to the nodegroup, and then configure VS2 as the backup for VS1.
  - Assume you have a content switching virtual server, CSVS1, whose target load balancing virtual server is LBVS1. To add CSVS1 to a nodegroup, first remove LBVS1 as the target. Then, bind each server individually to the nodegroup, and then configure LBVS1 as the target.
  - Assume you have a load balancing virtual server, LBVS1, that has a policy which invokes another load balancing virtual server, LBVS2. To add either one of the virtual servers, first remove the association. Then, bind each server individually to the nodegroup, and then reassociate the virtual servers.
- You cannot bind an entity to a nodegroup that has no nodes and that has the strict option enabled. Consequently, you cannot unbind the last node of a nodegroup that has entities bound to it and that has the strict option enabled.
- The strict option cannot be modified for a nodegroup that has no nodes but has entities bound to it.

Backing up nodes in a nodegroup

By default, a nodegroup is designed to provide back up nodes for members of a nodegroup. If a nodegroup member goes down, a cluster node that is not a member of the nodegroup dynamically replaces the failed node. This node is called the replacement node.

Note: For a single-member nodegroup, a backup node is automatically preselected when an entity is bound to the nodegroup.

When the original member of the nodegroup comes up, the replacement node, by default, is replaced by the original member node.

From NetScaler 10.5 Build 50.10 onwards, however, the NetScaler appliance allows you to change this replacement behavior. When you enable the sticky option, the replacement node is retained even after the original member node comes up. The original node takes over only when the replacement node goes down.

You can also disable the backup functionality. To do this, you must enable the strict option. In this scenario, when a nodegroup member goes down, no other cluster node is picked up as a backup node. The original node continues being part of the nodegroup when it comes up. This option ensures that entities bound to a nodegroup are active only on nodegroup members.

> **Note**
>
> The strict and sticky option can be set only when creating a nodegroup.

Configuring nodegroups for spotted and partially-striped configurations

September 21, 2020

To configure a nodegroup for spotted and partially-striped configurations you must first create a nodegroup and then bind the required nodes to the nodegroup. You must then associate the required entities to that nodegroup. The entities that are bound to the nodegroup will be:

- **Spotted** - If bound to a nodegroup that has a single node.
- **Partially Striped** - If bound to a nodegroup that has more than one node.

Some points to remember:

- GSLB is supported on a cluster only when GSLB sites are bound to nodegroups that have a single cluster node. For more information, see Setting Up GSLB in a Cluster.
- NetScaler Gateway is supported on a cluster only when the VPN virtual servers are bound to nodegroups that have a single cluster node. The sticky option must be enabled on the nodegroup.
- For versions prior to NetScaler 11, application firewall is supported only on individual cluster nodes (spotted configuration). Application firewall profiles can be associated only with virtual

servers that are bound to nodegroups that have a single cluster node. This means that application You are not allowed to do the following:
  - Bind application firewall profiles to striped or partially striped virtual servers.
  - Bind the policy to a global bind point or to user-defined policy labels.
  - Unbind, from a nodegroup, a virtual server that has application firewall profiles.
- NetScaler 11 introduced application firewall support for striped and partially-striped configurations. For more information, see Application Firewall Support for Cluster Configurations.

Check NetScaler appliance Features Supported in a Cluster to see the NetScaler versions from which GSLB, Citrix Gateway, and application firewall are supported in a cluster.

To configure a nodegroup by using the command line interface

1. Log on to the cluster IP address.
2. Create a nodegroup. Type:

   “‘add cluster nodegroup -strict (**YES** | **NO**)

   ```
   **Example**
   ```

   add cluster nodegroup NG0 -strict YES
   ‘‘‘

3. Bind the required nodes to the nodegroup. Type the following command for each member of the nodegroup:

   “‘bind cluster nodegroup -node

   ```
   **Example**

    To bind nodes with IDs 1, 5, and 6.
   ```

   > bind cluster nodegroup NG0 -node 1
   > bind cluster nodegroup NG0 -node 5
   > bind cluster nodegroup NG0 -node 6
   > ‘‘‘

4. Bind the entity to the nodegroup. Type the following command once for every entity that you want to bind:

   “‘bind cluster nodegroup (-vServer | -identifierName | -gslbSite -service)

   ```
   > **Note**
   >
   > The gslbSite and service parameters are available from NetScaler 10.5 onwards.
   ```

```

**Example**

To bind virtual servers VS1 and VS2 and rate limit identifier
   named identifier1.
```

> bind cluster nodegroup NG0 -vServer VS1
>
> bind cluster nodegroup NG0 -vServer VS2
>
> bind cluster nodegroup NG0 -identifierName identifier1
>
> "'

5. Verify the configurations by viewing the details of the nodegroup. Type:

   "'show cluster nodegroup

```
**Example**
```

> show cluster nodegroup NG0
>
> "'

To configure a nodegroup by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Node Groups.**
3. In the details pane, click **Add.**
4. In the **Create Node Group** dialog box, configure the nodegroup:
   a) Under **Cluster Nodes**, click the **Add** button.
      - The Available list displays the nodes that you can bind to the nodegroup and the Configured list displays the nodes that are bound to the nodegroup.
      - Click the **+** sign in the Available list to bind the node. Similarly, click the **-** sign in the Configured list to unbind the node.

   b) Under **Virtual Servers**, select the tab corresponding to the type of virtual server that you want to bind to the nodegroup. Click the **Add** button.
      - The Available list displays the virtual servers that you can bind to the nodegroup and the Configured list displays the virtual servers that are bound to the nodegroup.
      - Click the **+** sign in the Available list to bind the virtual server. Similarly, click the - sign in the Configured list to unbind the virtual server.

Configuring redundancy for nodegroups

September 21, 2020

> **Note**
>
> Supported from NetScaler 10.5 Build 52.1115.e onwards.

Nodegroups can be configured such that when one nodegroup goes down, another nodegroup can take over and process traffic. For example, when a nodegroup NG1 goes down, NG2 takes over.

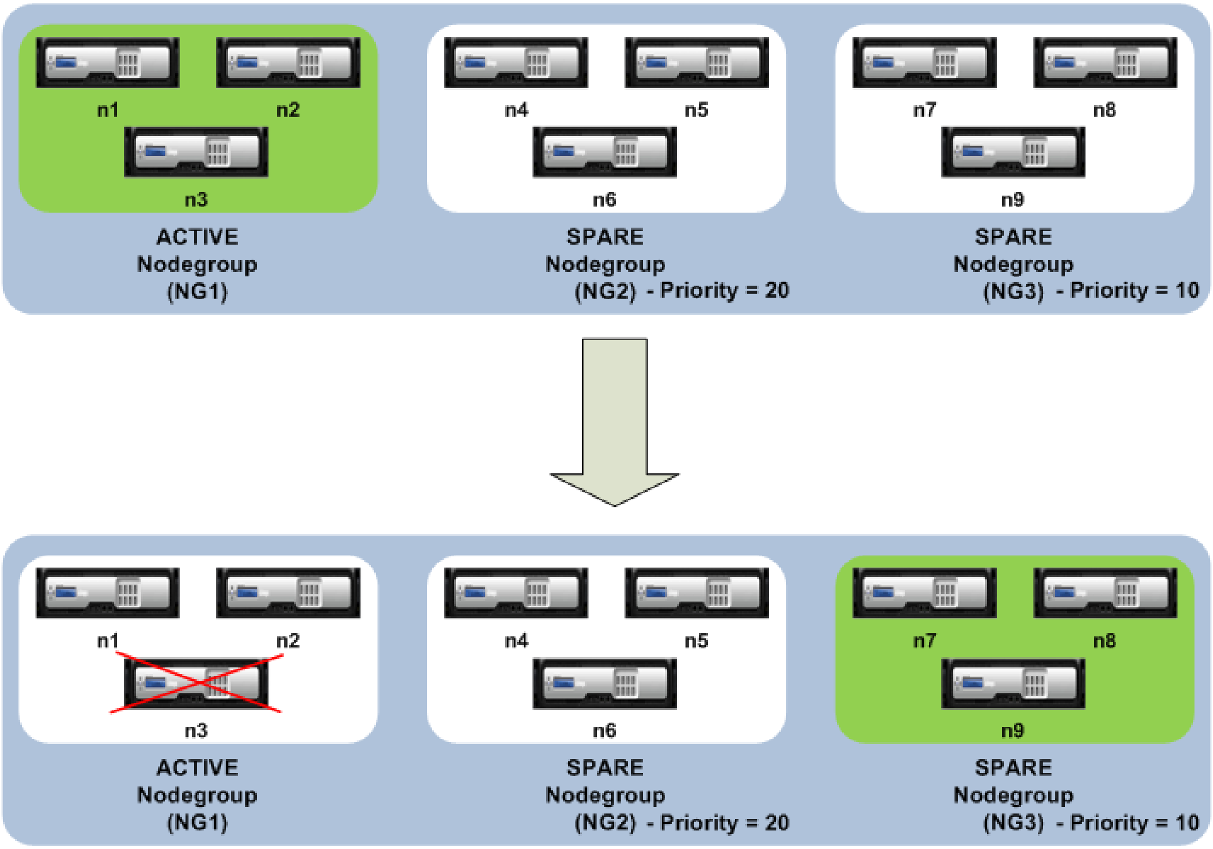
> **Note**
>
> This functionality can be used to configure datacenter redundancy where each nodegroup is configured as a datacenter.

To achieve this use case, cluster nodes must be logically grouped into nodegroups, where some nodegroups must be configured as ACTIVE and others as SPARE. The active nodegroup with the highest priority (that is, the lowest priority number) is made operationally active and therefore serves traffic. When a node from this operationally active nodegroup goes down, the node count of this nodegroup is compared with the node count of the other active nodegroups in order of their priority. If a nodegroup has a higher or equal node count, that nodegroup is made operationally active. Else, the spare nodegroups are checked.

> **Note**
>
> - Only one state-specific nodegroup can be active at a given point in time.
> - A cluster node inherits the state of the nodegroup. So, if a node with "SPARE" state is added to nodegroup with state as "ACTIVE", the node automatically behaves as an active node.
> - The preemption parameter that is defined for the cluster instance decides whether the initial active nodegroup will take control when the it comes up again.
> - A spare node group can take up a node group and host active traffic when an active node group goes down.

The following figure shows a nodegroup setup that has nodegroup redundancy defined. NG1 is initially the active nodegroup. When it loses one of the nodes, the spare nodegroup (NG3) with the highest priority starts serving traffic.

Figure 1. NetScaler cluster with nodegroup redundancy configured


n1
n2
n3
ACTIVE
Nodegroup
(NG1)
n4
n5
n6
SPARE
Nodegroup
(NG2) - Priority = 20
n7
n8
n9
SPARE
Nodegroup
(NG3) - Priority = 10
n1
n2
n3
ACTIVE
Nodegroup
(NG1)
n4
n5
n6
SPARE
Nodegroup
(NG2) - Priority = 20
n7
n8
n9
SPARE
Nodegroup
(NG3) - Priority = 10

Configuring redundancy for nodegroups

1. Log on to the cluster IP address.
2. Create the active nodegroup and bind the required cluster nodes.

```
    add cluster nodegroup NG1 -state ACTIVE
    bind cluster nodegroup NG1 -node n1
    bind cluster nodegroup NG1 -node n2
    bind cluster nodegroup NG1 -node n3
<!--NeedCopy-->
```

1. Create the spare nodegroup and bind the requisite nodes.

```
    add cluster nodegroup NG2 -state SPARE -priority 20
    bind cluster nodegroup NG2 -node n4
    bind cluster nodegroup NG2 -node n5
    bind cluster nodegroup NG2 -node n6
<!--NeedCopy-->
```

1. Create another spare nodegroup and bind the requisite nodes.

```
    add cluster nodegroup NG3 -state SPARE -priority 10
    bind cluster nodegroup NG3 -node n7
    bind cluster nodegroup NG3 -node n8
    bind cluster nodegroup NG3 -node n9
<!--NeedCopy-->
```

Disabling steering on the cluster backplane

September 21, 2020

> **Note**
>
> Supported from NetScaler 11 onwards.

The default behavior of a NetScaler cluster is to direct the traffic that it receives (flow receiver) to another node (flow processor) that must then process the traffic. This process of directing the traffic from flow receiver to flow processor occurs over the cluster backplane and is called steering.

If required, you can disable steering so that the process becomes local to the flow receiver and therefore makes the flow receiver as the flow processor. Such a configuration setup can come handy when you have a high latency link.

> **Note**
>
> This configuration is applicable only for striped virtual servers.

- For partially striped virtual servers, if the flow receiver is a non-owner node, the traffic is steered to a owner node. If however, the flow receiver is a owner node, then steering is disabled.
- For spotted virtual servers, flow receiver is the flow processor, and hence there is no need for steering.

Some points to remember when disabling the steering mechanism:

- Striped SNIPs are not supported as steering is disabled.
- MPTCP and FTP does not work.
- L2 mode must be disabled.
- If USIP is enabled, traffic may not reach back to the same node as the steering is disabled.
- Traffic that is directed to the cluster IP address is steered to the configuration coordinator.
- When a node joins or leaves a cluster, it is possible that more than 1/N connections will be affected. This is due to the fact that a change in the nodes available, may cause the routes to be rehashed. As a result, the traffic will be routed to another node and due to the non-availability of steering, the traffic will not be processed.

Steering can be disabled at the individual virtual server level or at the global level. The global configuration takes precedence over the virtual server setting.

- Disabling backplane steering for all striped virtual servers

  Configured at cluster instance level. Traffic meant for any striped virtual server will not be steered on cluster backplane.

  ```
  add cluster instance \<clId\> -processLocal ENABLED<!--NeedCopy-->
  ```

- Disabling backplane steering for a specific striped virtual server

  Configured on a striped virtual server. Traffic meant for the virtual server will not be steered on cluster backplane.

  ```
  add lb vserver <name> <serviceType> -processLocal ENABLED<!--NeedCopy
  -->
  ```

Synchronizing cluster configurations

September 21, 2020

NetScaler appliance configurations that are available on the configuration coordinator are synchronized to the other nodes of the cluster when:

- A node joins the cluster
- A node rejoins the cluster
- A new command is executed through the cluster IP address.

Additionally, you can forcefully synchronize the configurations that are available on the configuration coordinator (full synchronization) to a specific cluster node. Make sure you synchronize one cluster node at a time, otherwise the cluster can get affected.

To synchronize cluster configurations by using the command line interface

At the command prompt of the appliance on which you want to synchronize the configurations, type:

force cluster sync

To synchronize cluster configurations by using the configuration utility

1. Log on to the appliance on which you want to synchronize the configurations.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Utilities**, click Force cluster sync.
4. Click **OK.**

Synchronizing time across cluster nodes

September 21, 2020

The cluster uses Precision Time Protocol (PTP) to synchronize the time across cluster nodes. PTP uses multicast packets to synchronize the time. If there are some issues in time synchronization, you must disable PTP and configure Network Time Protocol (NTP) on the cluster.

To enable/disable PTP by using the command line interface

At the command prompt of the cluster IP address, type:

```
set ptp -state disable
```

To enable/disable PTP by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Utilities**, click **Configure PTP Settings.**
4. In the **Enable/Disable PTP** dialog box, select whether you want to enable or disable PTP.
5. Click **OK.**

Synchronizing cluster files

September 21, 2020

The files available on the configuration coordinator are called cluster files. These files are automatically synchronized on the other cluster nodes when the node is added to the cluster and periodically, during the lifetime of the cluster. Additionally, you can manually synchronize the cluster files.

The directories and files from the configuration coordinator that are synchronized are:

- /nsconfig/ssl/
- /var/netscaler/ssl/
- /var/vpn/bookmark/
- /nsconfig/dns/
- /nsconfig/htmlinjection/
- /netscaler/htmlinjection/ens/
- /nsconfig/monitors/
- /nsconfig/nstemplates/

- /nsconfig/ssh/
- /nsconfig/rc.netscaler
- /nsconfig/resolv.conf
- /nsconfig/inetd.conf
- /nsconfig/syslog.conf
- /nsconfig/snmpd.conf
- /nsconfig/ntp.conf
- /nsconfig/httpd.conf
- /nsconfig/sshd_config
- /nsconfig/hosts
- /nsconfig/enckey
- /var/nslw.bin/etc/krb5.conf
- /var/nslw.bin/etc/krb5.keytab
- /var/lib/likewise/db/
- /var/download/
- /var/wi/tomcat/webapps/
- /var/wi/tomcat/conf/Catalina/localhost/
- /var/wi/java_home/lib/security/cacerts
- /var/wi/java_home/jre/lib/security/cacerts
- /nsconfig/license/
- /nsconfig/rc.conf

> **Tip**
>
> Files (certificates and key files) that are copied to the cluster configuration coordinator manually (or through the shell) are not automatically available on the other cluster nodes. You must execute the "sync cluster files" command from the cluster IP address before executing a command that depends on these file(s).

To synchronize cluster files by using the command line interface

At the command prompt of the cluster IP address, type:

```
sync cluster files <mode><!--NeedCopy-->
```

To synchronize cluster files by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, under **Utilities**, click Synchronize cluster files.
4. In the **Synchronize** cluster files dialog box, select the files to be synchronized in the Mode drop-down box.

5. Click **OK.**

Viewing the statistics of a cluster

September 21, 2020

You can view the statistics of a cluster instance and cluster nodes to evaluate the performance or to troubleshoot the operation of the cluster.

To view the statistics of a cluster instance by using the command line interface

At the command prompt of the cluster IP address, type:

```
stat cluster instance <clId>
```

To view the statistics of a cluster node by using the command line interface

At the command prompt of the cluster IP address, type:

```
stat cluster node <nodeid>
```

> **Note**
>
> When executed from the cluster IP address, this command displays the cluster level statistics. However, when executed from the NSIP address of a cluster node, the command displays node level statistics.

To view the statistics of a cluster instance by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster.**
3. In the details pane, in the center of the page, click **Statistics.**

To view the statistics of a cluster node by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, select a node and click **Statistics** to view the statistics of the node. To view the statistics of all the nodes, click **Statistics** without selecting a specific node.

Discovering NetScaler appliances

September 21, 2020

You can discover the appliances present in the same subnet as the current node. The required discovered appliances can then be selectively added to the cluster. This operation can be performed to either create a new cluster or to add nodes to an existing cluster.

> **Note**
>
> - The discover operation can be performed only through the configuration utility.
> - This operation cannot discover NetScaler appliances from different networks.
> - When performing this operation to add nodes to an existing cluster, the L3 VLAN configurations will be cleared from the node. You must make sure to define these configurations once the appliance is added to the cluster.

To discover appliances by using the configuration utility

1. Log on to the cluster IP address.
2. Navigate to **System > Cluster > Nodes.**
3. In the details pane, at the bottom of the page, click **Discover NetScalers.**
4. In the **Discover NetScalers** dialog box, set the following parameters:
   - **IP address range** - Specify the range of IP addresses within which you want to discover appliances. For example, you can search for all NSIP addresses between 10.102.29.4 to 10.102.29.15 by specifying this option as 10.102.29.4 - 15.
   - **Backplane interface** - Specify the interfaces to be used as the backplane interface. This is an optional parameter. If you do not specify this parameter, you must update it after the node is added to the cluster.
5. Click **OK.**
6. Select the appliances that you want to add to the cluster.
7. Click **OK.**

Disabling a cluster node

September 21, 2020

You can temporarily remove a node from a cluster by disabling the cluster instance on that node. A disabled node is not synchronized with the cluster configurations. When the node is enabled again, the cluster configurations are automatically synchronized on it. For more information, see Synchronization across cluster nodes.

A disabled node cannot serve traffic and all existing connections on this node are terminated.

> **Note**
>
> If the configurations of a disabled non-configuration coordinator node are modified (through the NSIP address of the node), the configurations are not automatically synchronized on that node. You must manually synchronize the configurations as described in Synchronizing Cluster Configurations.

To disable a cluster node by using the command line interface

At the command prompt of the node that you want to disable, type:

```
disable cluster instance <clId><!--NeedCopy-->
```

> **Note**
>
> To disable the cluster, run the disable cluster instance command on the cluster IP address.

To disable a cluster node by using the configuration utility

1. On the node that you want to disable, navigate to **System > Cluster**, and click **Manage Cluster.**
2. In the **Configure** cluster instance dialog box, unselect the **Enable** cluster instance check box.

   > **Note**
   >
   > To disable the cluster instance on all the nodes, perform the above procedure on the cluster IP address.

Removing a cluster node

September 21, 2020

When a node is removed from the cluster, the cluster configurations are cleared from the node (by internally executing the clear ns config -extended command). The SNIP addresses, MTU settings of the backplane interface, and all VLAN configurations (except the default VLAN and NSVLAN) are also cleared from the appliance.

> **Note**
>
> - If the deleted node was the cluster configuration coordinator, another node is automatically selected as the cluster configuration coordinator, and the cluster IP address is assigned to that node. All the current cluster IP address sessions will be invalid and you will have to

> start a new session.
> - To delete the whole cluster, you must remove each node individually. When you remove the last node, the cluster IP address(es) are deleted.
> - When an active node is removed, the traffic serving capability of the cluster is reduced by one node. Existing connections on this node are terminated.

To remove a cluster node by using the command line interface

For NetScaler 10.1 and later versions

Logon to the cluster IP address and at the command prompt, type:

`rm cluster node <nodeId><!--NeedCopy-->`

> **Note**
>
> If the cluster IP address is unreachable from the node, execute the rm cluster instance command on the NSIP address of that node itself.

For NetScaler 10

1. Log on to the node that you want to remove from the cluster and remove the reference to the cluster instance.

   ```
   rm cluster instance <clId>

   save ns config
   ```

2. Log on to the cluster IP address and remove the node from which you removed the cluster instance.

   ```
   rm cluster node <nodeId>

   save ns config
   ```

   Make sure you do not run the `rm cluster node<!--NeedCopy-->` command from the local node as this results in inconsistent configurations between the configuration coordinator and the node.

To remove a cluster node by using the configuration utility

On the cluster IP address, navigate to **System > Cluster > Nodes**, select the node you want to remove and click **Remove.**

Removing a node from a cluster deployed using cluster link aggregation

September 21, 2020

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism, you must make sure that the node is made passive so that it does not receive any traffic and then, on the upstream switch, remove the corresponding interface from the channel.

For detailed information on cluster link aggregation, see Using Cluster Link Aggregation.

To remove a node from a cluster that uses cluster link aggregation as the traffic distribution mechanism

1. Log on to the cluster IP address.
2. Set the state of the cluster node that you want to remove to PASSIVE.

   ```
   set cluster node <nodeId> -state PASSIVE<!--NeedCopy-->
   ```

3. On the upstream switch, remove the corresponding interface from the channel by using switch-specific commands.

   > **Note**
   >
   > You do not have to manually remove the nodes interface on the cluster link aggregation channel. It is automatically removed when the node is deleted in the next step.

4. Remove the node from the cluster.

   ```
   rm cluster node <nodeId><!--NeedCopy-->
   ```

Detecting jumbo probe on a cluster

September 21, 2020

If a Jumbo frame is enabled on a Cluster interface, the backplane interface should be large enough to support the all packets in the Jumbo frame. This is achieved by setting the Maximum Transmission Unit (MTU) of the backplane as:

Backplane_MTU = maximum (all cluster interface MTU’s) + 78

To verify the above configuration, you must send a jumbo probe (of the above computational size) to all peer nodes of a Cluster setup. If the probe does not succeed, the appliance displays a warning message in the output of the “show cluster instance” command.

In the command interface mode, type the following command:

```
> show cluster instance
Cluster ID: 1
Dead Interval: 3 secs
Hello Interval: 200 msecs
Preemption: DISABLED
Propagation: ENABLED
Quorum Type: MAJORITY
INC State: DISABLED
Process Local: DISABLED
Cluster Status: ENABLED(admin), ENABLED(operational), UP
```

> **Warning**
>
> The MTU for a backplane interface must be large enough to handle all packets in the frame. It must be equal to <MTU_VAL>. If the recommended value is not user configurable, you must review the MTU value of jumbo interfaces.
>
> | Sl. no | Member Nodes | Health | Admin State | Operation State |
> |---|---|---|---|---|
> | 1 | Node ID: 1; Node IP: 10.102.53.167 | UP | Active | ACTIVE (Configuration Coordinator) |
> | 2 | Node ID: 2; Node IP: 10.102.53.168 | UP | Active | Active |

Route monitoring for dynamic routes in cluster

September 21, 2020

You can use a route monitor to make a Cluster node dependent on the internal routing table whether it contains or does not contain dynamically learnt route. In a Cluster system, a route monitor on each node checks the internal routing table to ensure there is a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

In a cluster deployment, if the client-side or server side-link of a node goes down, traffic is steered to this node through the peer nodes for processing. The steering of traffic is implemented by configuring dynamic routing and adding static ARP entries, pointing to the special MAC address of each node, on all the nodes. If there are a large number of nodes in a cluster deployment, adding and managing static ARP entries with special MAC addresses on all the nodes is a cumbersome task. Now, nodes implicitly use special MAC addresses for steering packets. Therefore, static ARP entries pointing to special MAC addresses are no longer required to be added to the cluster nodes.

To bind a cluster node using the command line interface

At the command prompt, type:

```
bind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
unbind cluster node <nodeId> (-routeMonitor <ip_addr|ipv6_addr|*> [<
    netmask>])
```

Consider a scenario where Node 1 is bound to route monitor 1.1.1.0 255.255.255.0. When a dynamic route fails, Node 1 become INACTIVE. Health status is available in the Show Cluster Node command by node id as show below.

```
Node ID: 1
IP:  10.102.169.96
Backplane:  1/1/2
Health: NOT UP
Reason(s): Route Monitor(s) of the node have failed
Route Monitor -  Network: 1.1.1.0   Netmask: 255.255.255.0   State:
    DOWN
```

Monitoring cluster setup using SNMP MIB with SNMP link

September 21, 2020

SNMP mib is device specific information that has been configured on the SNMP agent for the purpose of identifying a NetScaler appliance, such as appliance name, administrator, location. In a cluster setup, you can now configure the SNMP MIB in any node by including the "ownerNode" parameter in the set snmp mib command. Without this parameter, the set snmp mib command applies only to the Cluster Coordinator (CCO) node.

To display the MIB configuration for a cluster node other than the CCO, include the "ownerNode" parameter in the show snmp mib command

Configuring SNMP MIB on CLIP

To configure and view MIB configuration on CLIP by using the command line interface.

```
set snmp mib [-contact <string>] [-name <string>] [-location <string>]
       [-customID <string>] [-ownerNode <positive_integer>]
Done
show snmp mib [-ownerNode <positive_integer>]

 > set mib -contact John -name NS59 -location San Jose -customID 123 -
    ownerNode 3
```

```
Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun  7 2016, 10:27:29
        sysUpTime:   124300
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  John
        sysName:     NS59
        sysLocation: San Jose
        sysServices: 72
        Custom ID: 123
 Done

> unset mib -contact -name -location -customID -ownerNode 3
 Done
> sh mib -ownerNode 3
        --------------------
         Cluster Node ID: 3
        --------------------
        NetScaler system MIB:
        sysDescr:    NetScaler NS11.1: Build 46.4.a.nc, Date: Jun 7 2016, 10:27:29
        sysUpTime:   146023
        sysObjectID: .1.3.6.1.4.1.5951.1.1
        sysContact:  WebMaster (default)
        sysName:     NetScaler
        sysLocation: POP (default)
        sysServices: 72
        Custom ID: Default
 Done
```

Monitoring command propagation failures in a cluster deployment

September 21, 2020

In a cluster deployment of NetScaler appliances, you can use the new command "show prop status" for faster monitoring and troubleshooting of issues related to command-propagation failure on non-CCO nodes. This command displays up to 20 of the most recent command propagation failures on all non-CCO nodes. You can use either the NetScaler appliance command line or the NetScaler GUI to perform this operation after accessing them through the CLIP address or through the NSIP address of any node in the cluster deployment.

IPv6 ready logo support for clusters

September 21, 2020

You can test clustered appliances for IPv6 Ready Logo certification. Modified commands for testing IPv6 core protocols, such as for ND test cases, Router Solicitation processing, and sending route advertisement and router redirection messages are available in a clustered setup. Following are the IPv6 functionalities available for testing the IPv6 core protocols.

Following are the modified functionalities available to pass IPv6 core protocols, such as ND test cases, Router Solicitation processing and sending Route advertisement and router redirection messaging in IPv6readylogo phase2 test suite.

- Link Local SNIPs
- Address Resolution and Neighbor Unreachability
- Router and Prefix Discovery
- Router Redirection
- DoDAD

With these modified commands, the following configurations are supported in a clustered appliance.

Supportable configurations for testing IPv6 core protocols

For a clustered setup to pass IPv6 Ready Logo test cases, you can execute the following configurations on the Cluster Management IP address (CLIP).

- global IP6 configuration
- basic IPv6 configuration
- additional IPv6 configurations

Global configuration

A global IPv6 configuration enables you to set the global IPv6 parameters (such as relearning, routeRedirection, ndBasereachTime, nRetransmissionTime, natprefix, td, and doodad) to run the basic IPv6 configuration.

At the command prompt, type the following:

```
set ipv6 [-ralearning ( ENABLED | DISABLED )] [-routerRedirection (
    ENABLED | DISABLED )] [-ndBasereachTime<positive_integer>][-
    ndRetransmissionTime <positive_integer>] [-natprefix <ipv6_addr|*>[-
    td<positive_integer>]] [-doDAD ( ENABLED | DISABLED )]
```

Basic IPv6 configuration

The basic IPv6 configuration enables you to create an IPv6 address and bind to a VLAN interface. You must perform the following configurations to test the IPv6 core protocols.

To add a VLAN to the clustered setup by using the NetScaler command line

At the command prompt, type:

```
 add vlan <id>
```

To add another VLAN to the clustered set up by using the NetScaler command line

At the command prompt, type:

```
 add vlan <id>
```

To bind an interface to a VLAN by using the NetScaler command line

At the command prompt, type:

```
bind vlan <id> -ifnum <interface_name>
```

To bind an interface to a VLAN by using the NetScaler command line

This command adds the global prefix as on-link prefix into RA information for subsequent Router Advertisements. At the command prompt, type:

```
bind vlan <id> -ifnum <interface_name>
```

To add IPv6 SNIP address on a VLAN by using the NetScaler command line

At the command prompt, type the following:

```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

To add IPv6 address on VLAN by using the NetScaler command line

At the command prompt, type the following:

```
add ns ip6 <IPv6Address>@ [-scope ( global | link-local )][-type <type>
```

To bind IPv6 address to VLAN by using the NetScaler command line

At the command prompt, type the following:

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

To bind IPv6 address to VLAN by using the NetScaler command line

At the command prompt, type the following:

```
bind vlan <id> [-ifnum <interface_name> [-tagged]][-IPAddress <ip_addr|
    ipv6_addr|
```

To display link local IPv6 address attached to VLAN by using the NetScaler command line

At the command prompt, type the following:

```
sh VLAN
```

Example 1

```
add vlan 2
add vlan 3
bind vlan 2 -ifnum 1/2
bind vlan 3 -ifnum 1/3
add ip6 fe80::9404:60ff:fedd:a464/64 -vlan 2 -scope link-local -type
    SNIP
add ip6 fe80::c0ee:7bff:fede:263f/64 -vlan 3 -scope link-local -type
    SNIP
add ip6 3ffe:501:ffff:100:9404:60ff:fedd:a464/64 -vlan 2
add ip6 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64 -vlan 3
bind vlan 2 -ipAddress 3ffe:501:ffff:100:9404:60ff:fedd:a464/64
bind vlan 3 -ipAddress 3ffe:501:ffff:101:c0ee:7bff:fede:263f/64
```

Example 2

```
sh vlan
1)      VLAN ID: 2      VLAN Alias Name:
        Interfaces : 1/6
        IPs :
             3ffe:501:ffff:100:2e0:edff:fe15:ea2a/64
3)      VLAN ID: 3      VLAN Alias Name:
        Link-local IPv6 addr: fe80::9404:60ff:fedd:a464/64
        Interfaces : 1/5
        IPs :
             3ffe:501:ffff:101:2e0:edff:fe15:ea2b/64
 Done
```

Additional IPv6 cluster configuration

To test IPv6 core protocols, you can use the following new or modified IPv6 configurations.

To set VLAN specific Router Advertisement parameters by using the NetScaler command line

At the command prompt, type:

```
set nd6RAvariables -vlan <positive_integer> [-ceaseRouterAdv ( YES | NO )] [-sendRouterAdv ( YES | NO )] [-srcLinkLayerAddrOption ( YES | NO )][-onlyUnicastRtAdvResponse ( YES | NO )] [-managedAddrConfig ( YES | NO)] [-otherAddrConfig ( YES | NO )] [-currHopLimit <positive_integer>][-maxRtAdvInterval <positive_integer>] [-minRtAdvInterval<positive_integer>] [-linkMTU <positive_integer>] [-reachableTime<positive_integer>] [-retransTime <positive_integer>] [-defaultLifeTime<integer>]
```

To set an on-link global prefix's configurable parameters by using the NetScaler command line

At the command prompt, type:

```
set onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-decrementPrefixLifeTimes ( YES | NO )] [-prefixValideLifeTime   <positive_integer>] [-prefixPreferredLifeTime <positive_integer>]
```

To add configurable parameters to an on-link global prefix by using the NetScaler command line

At the command prompt, type:

```
add onLinkIPv6Prefix <ipv6Prefix> [-onlinkPrefix ( YES | NO )][-autonomusPrefix ( YES | NO )] [-depricatePrefix ( YES | NO )][-decrementPrefixLifeTimes ( YES | NO )]-prefixValideLifeTime <positive_integer>][-prefixPreferredLifeTime <positive_integer>]
```

To set on-link link IPv6 prefix's configurable parameters by using the NetScaler command line

At the command prompt, type the following:

```
help set onLinkIPv6Prefix
```

To bind an on-link link IPv6 prefix's configurable parameters by using the NetScaler command line

At the command prompt, type:

```
help bind nd6RAvariables
```

To show nd6RAvariables by using the NetScaler command line

At the command prompt, type:

```
help sh nd6RAvariables
```

Example

```
> sh nd6RAvariables
  1) Vlan : 1
     SendAdvert     : NO   CeaseAdv       : NO        SourceLLAddress: YES
     UnicastOnly    : NO   ManagedFlag    : NO        OtherConfigFlag: NO
     CurHopLimit    : 64   MaxRtrAdvInterv: 600       MinRtrAdvInterv: 198
     LinkMTU        : 0    ReachableTime  : 0          RetransTimer   : 0
     DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    : 0

  2) Vlan : 2
     SendAdvert     : NO   CeaseAdv       : NO         SourceLLAddress: YES
     UnicastOnly    : NO   ManagedFlag    : NO         OtherConfigFlag: NO
     CurHopLimit    : 64   MaxRtrAdvInterv: 600        MinRtrAdvInterv: 198
     LinkMTU        : 0    ReachableTime  : 0          RetransTimer   : 0
     DefaultLifetime: 1800 LastRAsentTime : 0          NextRAdelay    : 0
Done
>
> sh nd6Ravariables  - vlan 2
  1) Vlan : 2
```

```
        SendAdvert     : NO   CeaseAdv       : NO          SourceLLAddress: YES
        UnicastOnly    : NO   ManagedFlag    : NO          OtherConfigFlag: NO
        CurHopLimit    : 64   MaxRtrAdvInterv: 600         MinRtrAdvInterv: 198
        LinkMTU        : 0    ReachableTime  : 0           RetransTimer   : 0
        DefaultLifetime: 1800 LastRAsentTime : 0           NextRAdelay    : 0
        Prefix :
3ffe:501:ffff:100::/64
Done
```

Managing cluster heartbeat messages

September 21, 2020

Managing heartbeat messages in a cluster is similar to managing them in a high availability (HA) configuration. Nodes can send and receive heartbeat messages to and from each other on all interfaces that are enabled. To avoid increased traffic resulting from hearbeat messages, you can now disable the heartbeat option on node interfaces. However, the heartbeat option on the backplane interface cannot be disabled, because it is required for maintaining connectivity among the cluster nodes.

For more information about managing heart messages, see Managing High Availability Heartbeat Messages on a NetScaler Appliance.

To manage heartbeat messages on a node interface by using the NetScaler CLI

At the command prompt, type:

```
set interface <ID> [-HAHeartBeat (ON | OFF)]
Show interface <ID>
```

Configuring owner node response status

September 21, 2020

In a cluster setup, on a node has a spotted SNIP address, you can configure the ownerDownResponse option. By default, the option is enabled, allowing the spotted IP address to respond to PING or ARP requests (coming from the upstream router) when the node is inactive. If you disable the option, the IP address cannot respond to the router requests when the owner node is inactive.

To know how this feature is used for monitoring static routes in ECMP deployment, see Using Equal Cost Multiple Path (ECMP) topic.

To set owner node response status by using the NetScaler CLI

At the command prompt, type:

```
add ns ip <IPAddress> [-ownerNode <positive_integer>] [-
    ownerDownResponse (YES | NO )] [-td <positive_integer>]
```

Example

```
add ns ip 2.2.2.2 255.255.255.0 -ownernode 6  - ownerdownResponse YES
```

To set owner node response status by using the NetScaler GUI

1. Navigate to **System > Network > IPs** and click **Add** to create a spotted SNIP address.
2. On the **Create IP Address** page, select or clear the **ownerDownResponse** check box.

To edit owner node response status by using the NetScaler GUI

Navigate to **System > Network > IPs**, select an IP address, and click **Edit** to select or clear the **ownerDownResponse** check box.

Monitor static route (MSR) support for inactive nodes in a spotted cluster configuration

September 21, 2020

In a cluster set up with MSR option enabled on the route, only active nodes can probe to a static route and reach a network while inactive and spare nodes have no link to the route and cannot probe to it. You can now configure an inactive or spare node to send PING and ARP probe to IPv4 route and send ping6 and nd6 probe to IPv6 route. You can perform this only in a Spotted Cluster configuration in which the SNIP address is active and owned exclusively only by one node.

VRRP interface binding in a single node active cluster

September 21, 2020

When you migrate a high availability (HA) setup to a cluster setup, all configurations must be compatible and must be supportable in the cluster. To achieve this, you can now configure virtual router IDs (VRIDs and VRID6s) on a node interface.

Important: Currently, only a single-node active cluster system supports VRIDs and VRID6s.

For instructions for configuring VRIDs and VRID6s, see Configuring Virtual MAC Addresses.

To configure a virtual router ID on a single-node active cluster, add the VRID or VRID6 and bind it to the cluster-node interface.

To add a VRID by using the NetScaler CLI

At the command prompt, type:

```
add vrID <ID>
```

To bind a VRID to the cluster-node interface by using the NetScaler CLI

At the command prompt, type:

```
Bind vrid <ID> -ifnum <interface_name> | -trackifNum <interface_name

Add vrID 100
Bind vrid 100  - ifnum 1/1  1/2
done
```

To add a VRID6 by using the NetScaler CLI

At the command prompt, type:

```
add vrID6 <ID>
```

To bind a VRID6 to cluster node interface by using the NetScaler CLI

At the command prompt, type:

```
bind vrid6 <ID> -ifnum <interface_name> | -trackifNum <interface_name>

Add vrID6 100
Bind vrid6 100  - ifnum 1/1  1/2
Done
```

Cluster setup and usage scenarios

September 21, 2020

This section explains some scenarios in which the NetScaler cluster can be set up, and configured for different features and network topologies. Provide feedback if you want any other scenarios to be documented.

Creating a two-node cluster

September 21, 2020

A two-node cluster is an exception to the rule that a cluster is functional only when a minimum of (n/2 +1) nodes, where n is the number of cluster nodes, are able to serve traffic. If that formula were applied to a two-node cluster, the cluster would fail if one node went down (n/2 +1=2).

A two-node cluster is functional even if only one node is able to serve traffic.

Creating a two node cluster is the same as creating any other cluster. You must add one node as the configuration coordinator and the other node as the other cluster node.

> **Note**
>
> Incremental configuration synchronization is not supported in a two-node cluster. Only full synchronization is supported.

Migrating an HA setup to a cluster setup

September 21, 2020

Migrating an existing high availability (HA) setup to a cluster setup requires you to first remove the NetScaler appliances from the HA setup and create a backup of the HA configuration file. You must then use the two appliances to create a cluster and upload the backed-up configuration file to the cluster.

> **Note**
>
> - Before uploading the backed-up HA configuration file to the cluster, you must modify it to make it cluster compatible. Refer to the relevant step of the procedure below.
> - Use the **batch -f <backup_filename>** command to upload the backed-up configuration file.

The above approach is a very basic migration solution which results in downtime for the deployed application. As such, it must be used only in deployments where there is no consideration to application availability.

However, in most deployments, the availability of the application is of paramount importance. For such cases, you must use the approach where an HA setup can be migrated to a cluster setup without any resulting downtime. In this approach, an existing HA setup is migrated to a cluster setup by first removing the secondary appliance and using that appliance to create a single-node cluster. After the cluster becomes operational and serves traffic, the primary appliance of the HA setup is added to the cluster.

To convert a HA setup to cluster setup (without any downtime) by using the command line interface

Let us consider the example of a HA setup with primary appliance (NS1) - 10.102.97.131 and secondary appliance (NS2) - 10.102.97.132.

1. Make sure the configurations of the HA pair are stable.
2. Log on to any one of the HA appliances, go to the shell, and create a copy of the ns.conf file (for example, ns_backup.conf).
3. Log on to the secondary appliance, NS2, and clear the configurations. This operation removes NS2 from the HA setup and makes it a standalone appliance.

   ```
   > clear ns config full
   <!--NeedCopy-->
   ```

   > **Note**
   >
   > - This step is required to make sure that NS2 does not start owning VIP addresses, now that it is a standalone appliance.
   > - At this stage, the primary appliance, NS1, is still active and continues to serve traffic.

4. Create a cluster on NS2 (now no longer a secondary appliance) and configure it as a PASSIVE node.

   ```
   > add cluster instance 1

   > add cluster node 0 10.102.97.132 -state PASSIVE -backplane 0/1/1

   > add ns ip 10.102.97.133 255.255.255.255 -type CLIP

   > enable cluster instance 1

   > save ns config
   ```

```

> reboot -warm
<!--NeedCopy-->
```

5. Modify the backed-up configuration file as follows:

   - Remove the features that are not supported on a cluster. For the list of unsupported features, see NetScaler appliance Features Supported by a Cluster. This is an optional step. If you do not perform this step, the execution of unsupported commands will fail.

   - Remove the configurations that have interfaces, or update the interface names from the c/u convention to the n/c/u convention.

     Example

     ```
     > add vlan 10 -ifnum 0/1

     should be changed to

     > add vlan 10 -ifnum 0/0/1 1/0/1
     <!--NeedCopy-->
     ```

   - The backup configuration file can have SNIP addresses. These addresses are striped on all the cluster nodes. It is recommended that you add spotted IP addresses for each node.

     Example

     ```
     > add ns ip 1.1.1.1 255.255.255.0 -ownerNode 0

     > add ns ip 1.1.1.2 255.255.255.0 -ownerNode 1
     <!--NeedCopy-->
     ```

   - Update the hostname to specify the owner node.

     Example

     ```
     > set ns hostname ns0 -ownerNode 0

     > set ns hostname ns1 -ownerNode 1
     <!--NeedCopy-->
     ```

   - Change all other relevant networking configuration that depend on spotted IPs. For example, L3 VLAN, RNAT configuration which uses SNIPs as NATIP, INAT rules that refers to SNIPs/MIPs).

6. On the cluster, do the following:

- Make the topological changes to the cluster by connecting the cluster backplane, the cluster link aggregation channel, and so on.
- Apply configurations from the backed-up and modified configuration file to the configuration coordinator through the cluster IP address.

  ```
  > batch -f ns_backup.conf
  <!--NeedCopy-->
  ```

- Configure external traffic distribution mechanisms like ECMP or cluster link aggregation.

7. Switch the traffic from the HA setup to the cluster.
   - Log on to the primary appliance, NS1, and disable all the interfaces on it.

     ```
     > disable interface <interface_id>
     <!--NeedCopy-->
     ```

   - Log on to the cluster IP address and configure NS2 as an ACTIVE node.

     ```
     > set cluster node 0 -state ACTIVE
     <!--NeedCopy-->
     ```

> **Note**
>
> There might be a small amount (in the order of seconds) of downtime between disabling the interfaces and making the cluster node active.

8. Log on to the primary appliance, NS1, and remove it from the HA setup.
   - Clear all the configurations. This operation removes NS1 from the HA setup and makes it a standalone appliance.

     ```
     > clear ns config full
     <!--NeedCopy-->
     ```

   - Enable all the interfaces.

     ```
     > enable interface <interface_id>
     <!--NeedCopy-->
     ```

9. Add NS1 to the cluster.
   - Log on to the cluster IP address and add NS1 to the cluster.

     ```
     > add cluster node 1 10.102.97.131 -state PASSIVE -backplane 1/1/1
     <!--NeedCopy-->
     ```

- Log on to NS1 and join it to the cluster by sequentially executing the following commands:

  ```
  > join cluster -clip 10.102.97.133 -password nsroot

  > save ns config

  > reboot -warm
  <!--NeedCopy-->
  ```

10. Log on to NS1 and perform the required topological and configuration changes.
11. Log on to the cluster IP address and set NS1 as an ACTIVE node.

    ```
    > set cluster node 1 -state ACTIVE
    <!--NeedCopy-->
    ```

Transitioning between a L2 and L3 Cluster

September 21, 2020

> **Note**
>
> Supported from NetScaler 11 onwards.

An L2 cluster is one where all the nodes are from the same network and an L3 cluster is one that can include nodes from different networks. You can seamlessly transition from one type of cluster to the other without any downtime for the applications that are deployed on the NetScaler appliance.

Transitioning a cluster from L2 to L3

You can transition to an L3 cluster when you want the cluster to include nodes from other networks.

On the cluster IP address, do the following:

1. Create a nodegroup.

   Example

   ```
   > add cluster nodegroup NG0
   <!--NeedCopy-->
   ```

   This nodegroup is used in the next step to group all the nodes from the existing L2 cluster.

2. Transition the L2 cluster to an L3 cluster.

   Example

```
> set cluster instance 1 -inc ENABLED -nodegroup NG0
<!--NeedCopy-->
```

This command achieves the dual purpose of transitioning to L3 cluster and also adding all the nodes of the L2 cluster to the nodegroup.

3. Now, you can add more nodes to the cluster as explained in "Adding a Node to the Cluster".

Transitioning a cluster from L3 to L2

You can transition to an L2 cluster when you want to retain nodes that belong to a single network.

On the cluster IP address, do the following:

1. Remove the cluster nodes from the networks that you do not want to retain.

   Example

   ```
   > rm cluster node <nodeId>
   <!--NeedCopy-->
   ```

2. Transition the L3 cluster to a L2 cluster.

   Example

   ```
   > set cluster instance 1 -inc DISABLED
   <!--NeedCopy-->
   ```

   The cluster now includes nodes only of a single network.

Setting up GSLB in a cluster

September 21, 2020

> **Note**
>
> Supported from NetScaler 10.5 Build 52.11 onwards.

To set up GSLB in a cluster you must bind the different GSLB entities to a node group. The node group must have a single member node.

> **Note**
>
> - The parent-child topology of GSLB is not supported in a cluster.
> - If you have configured the static proximity GSLB method, make sure that the static proxim-

> ity database is present on all the cluster nodes. This happens by default if the database file is available at the default location. However, if the database file is maintained in a directory other than /var/netscaler/locdb/, you must manually synch the file to all the cluster nodes.

To set up GSLB in a cluster by using the command line interface:

Log on to the cluster IP address and perform the following operations at the command prompt:

1. Configure the different GSLB entities. For information, see GSLB Configuration Entities.

   > **Note**
   >
   > When creating the GSLB site, make sure that you specify the cluster IP address and public cluster IP address (needed only when the cluster is deployed behind a NAT device). While configuring a GSLB site, you must use the cluster IP address of the same site. These parameters are required to ensure the availability of the GSLB auto-sync functionality.

   ```
   add gslb site <siteName> <siteType> <siteIPAddress> -publicIP <ip_addr>
    -clip <ip_addr> <publicCLIP><!--NeedCopy-->
   ```

2. Create a cluster node group.

   ```
   add cluster nodegroup <name> [-sticky ( YES | NO )]<!--NeedCopy-->
   ```

   > **Note**
   >
   > Enable the sticky option if you want to set up GSLB based on VPN users.

3. Bind a single cluster node to the node group.

   ```
   bind cluster nodegroup <name> -node <nodeId><!--NeedCopy-->
   ```

4. Bind the local GSLB site to the nodegroup.

   ```
   bind cluster nodegroup <name> -gslbSite <string><!--NeedCopy-->
   ```

   > **Note**
   >
   > Make sure that the IP address of the local GSLB site IP address is striped (available across all cluster nodes).

5. Bind the ADNS (or ADNS-TCP) service or the DNS (or DNS-TCP) load balancing virtual server to the node group.

   To bind the ADNS service:

   “‘bind cluster nodegroup -service

   ~~~
   **To bind the DNS load balancing virtual server:**

   ```bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ~~~

6. Bind the GSLB virtual server to the node group.

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

7. [Optional] To setup GSLB based on VPN users, bind the VPN virtual server to the GSLB node group.

   ```
   bind cluster nodegroup <name> -vServer <string><!--NeedCopy-->
   ```

8. Verify the configurations.

   ```
   show gslb runningConfig<!--NeedCopy-->
   ```

To set up GSLB in a cluster by using the graphical user interface:

Log on to the cluster IP address and perform the following operations in the Configuration tab:

1. Configure the GSLB entities.

   Navigate to **Traffic Management > GSLB** to perform the required configurations.

2. Create a node group and perform other node group related configurations.

   Navigate to **System > Cluster > Node Groups** to perform the required configurations.

For the detailed configurations to be performed, see the description provided in the CLI procedure mentioned above.

Using cache redirection in a cluster

September 21, 2020

Cache redirection in a cluster works in the same way as it does on a standalone NetScaler appliance. The only difference is that the configurations are done on the cluster IP address. For more information on cache redirection, see "Cache Redirection."

Points to remember when using cache redirection in transparent mode on a cluster:

- Before configuring cache redirection, make sure that you have connected all nodes to the external switch and that you have linksets configured. Otherwise, client requests will be dropped.
- When MAC mode is enabled on a load balancing virtual server, make sure MBF mode is enabled on the cluster by using the enable ns mode MBF command. Otherwise, the requests are sent to origin server directly instead of being sent to the cache server.

Using L2 mode in a cluster setup

September 21, 2020

> **Note**
>
> Supported from NetScaler 10.5 and later releases.

To use L2 mode in a cluster setup, you must make sure of the following:

- Spotted IP addresses must be available on all the nodes as required.
- Linksets must be used to communicate with the external network.
- Asymmetric topologies or asymmetric cluster LA groups are not supported.
- Cluster LA group is recommended.
- Traffic is distributed between the cluster nodes only for deployments where services exist.

Using cluster LA channel with linksets

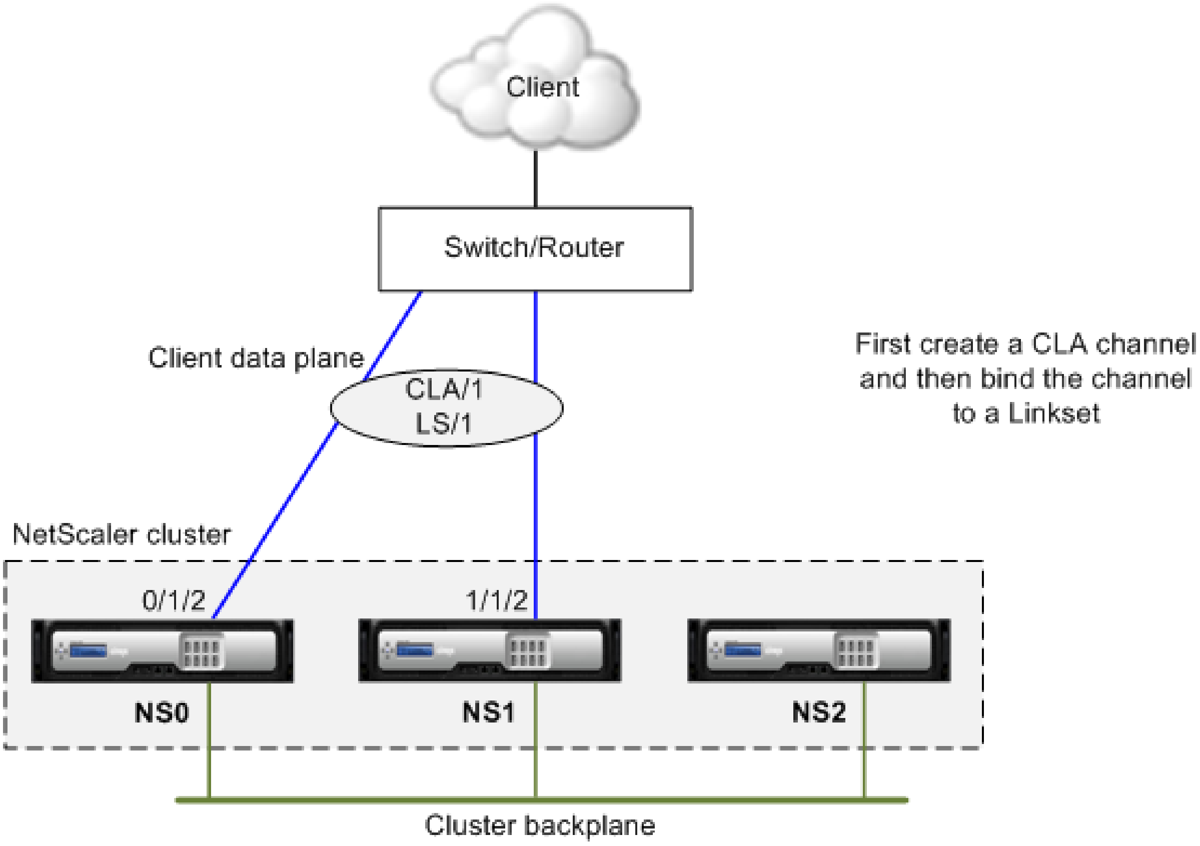
September 21, 2020

In an asymmetric cluster topology, some cluster nodes are not connected to the upstream network. In such a case, you must use linksets. To optimize the performance, you can bind the interfaces that are connected to the switch as a cluster LA channel and then bind the channel to a linkset.

To understand how a combination of cluster LA channel and linksets can be used, consider a three-node cluster for which the upstream switch has only two ports available. You can connect two of the cluster nodes to the switch and leave the other node unconnected.

> **Note**
>
> Similarly, you can also use a combination of ECMP and linksets in an asymmetric topology.

Figure 1. Linksets and cluster LA channel topology


Client
Switch/Router
Client data plane
CLA/1
LS/1
First create a CLA channel and then bind the channel to a Linkset
NetScaler cluster
0/1/2
1/1/2
NS0
NS1
NS2
Cluster backplane

To configure cluster LA channel and linksets by using the NetScaler appliance command line

1. Log on to the cluster IP address.
2. Bind the connected interfaces to a cluster LA channel.

```
add channel CLA/1  - ifnum 0/1/2 1/1/2
<!--NeedCopy-->
```

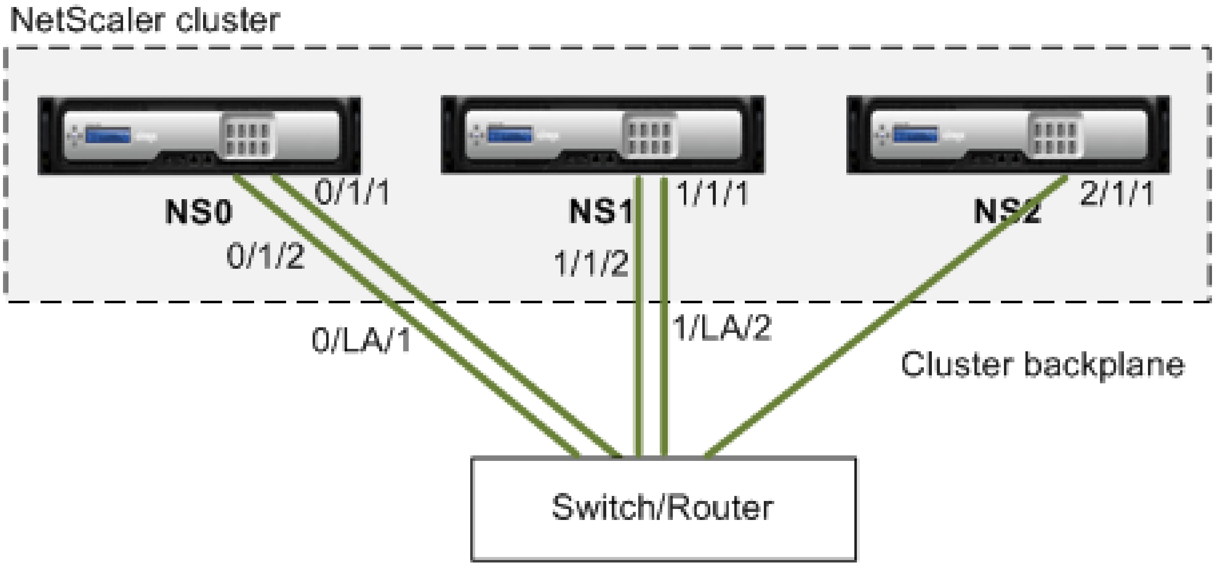
3. Bind the cluster LA channel to the linkset.

```
add linkset LS/1 -ifnum CLA/1
<!--NeedCopy-->
```

Backplane on LA channel

September 21, 2020

In this deployment, LA channels are used for the cluster backplane.


NetScaler cluster
NS0
0/1/1
0/1/2
NS1
1/1/1
1/1/2
NS2
2/1/1
0/LA/1
1/LA/2
Cluster backplane
Switch/Router

- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the backplane interfaces as LA channels

1. Create a cluster of nodes NS0, NS1, and NS2.

   a) Log on to the first node that you want to add to the cluster and do the following:

   ```
   create cluster instance 1
   add cluster node 0 10.102.29.60 -state ACTIVE
   enable cluster instance 1
   add ns ip 10.102.29.61 255.255.255.255 -type CLIP
   save ns config
   reboot -warm
   <!--NeedCopy-->
   ```

   b) Log on to the cluster IP address and do the following:

   ```
   add cluster node 1 10.102.29.70 -state ACTIVE
   add cluster node 2 10.102.29.80 -state ACTIVE
   <!--NeedCopy-->
   ```

   c) Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

   ```
   join cluster -clip 10.102.29.61 -password nsroot
   ```

```
save ns config
reboot -warm
<!--NeedCopy-->
```

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. Log on to the cluster IP address and do the following:

    a) Create the LA channels for nodes NS0 and NS1.

    ```
    add channel 0/LA/1 -ifnum 0/1/1 0/1/2
    add channel 1/LA/2 -ifnum 1/1/1 1/1/2
    <!--NeedCopy-->
    ```

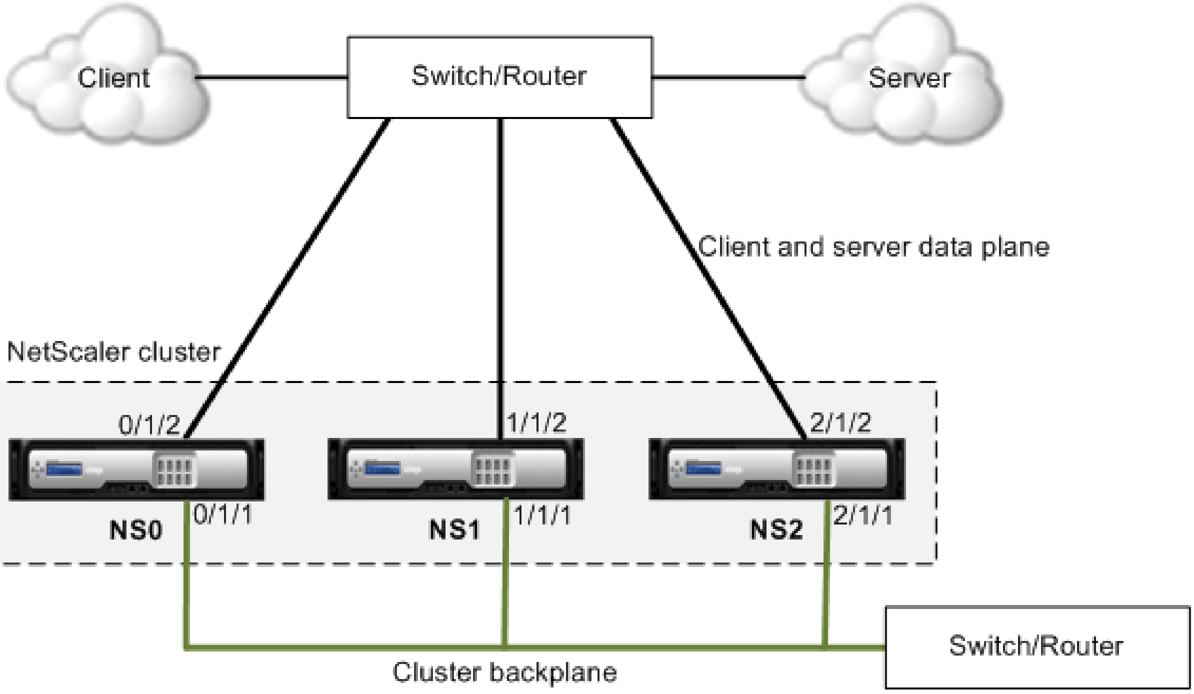
    b) Configure the backplane for the cluster nodes.

    ```
    set cluster node 0 -backplane 0/LA/1
    set cluster node 1 -backplane 1/LA/2
    set cluster node 2 -backplane 2/1/1
    <!--NeedCopy-->
    ```

Common interfaces for client and server and dedicated interfaces for backplane

September 21, 2020

This is a one-arm deployment of the NetScaler cluster. In this deployment, the client and server networks use the same interfaces to communicate with the cluster. The cluster backplane uses dedicated interfaces for inter-node communication.


Client
Switch/Router
Server
Client and server data plane
NetScaler cluster
0/1/2
1/1/2
2/1/2
NS0
NS1
NS2
0/1/1
1/1/1
2/1/1
Switch/Router
Cluster backplane

- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common interface for the client and server and a different interface for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
2. Log on to the first node that you want to add to the cluster and do the following:

```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

3. Log on to the cluster IP address and do the following:

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

4. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

1. On the cluster IP address, create VLANs for the backplane interfaces and for the client and server interfaces.

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1

//For the interfaces that are connected to the client and server
    networks.
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2
<!--NeedCopy-->
```

2. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

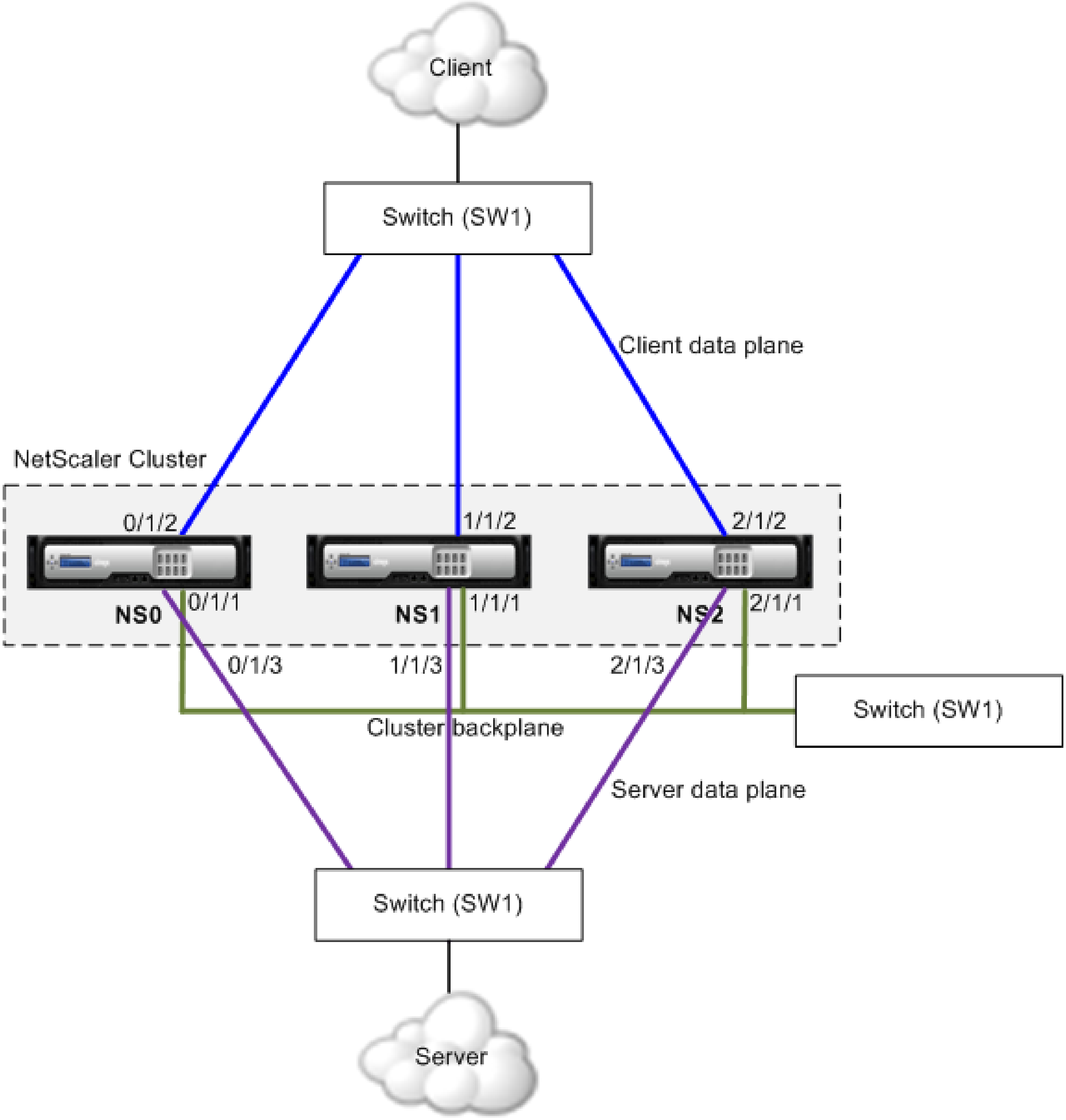
```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

//For the interfaces connected to the client and server networks.
    Repeat for each interface...
interface Ethernet2/47
switchport access vlan 200
switchport mode access
end
<!--NeedCopy-->
```

Common switch for client, server, and backplane

September 21, 2020

In this deployment, the client, server, and backplane use dedicated interfaces on the same switch to communicate with the NetScaler cluster.


Client
Switch (SW1)
Client data plane
NetScaler Cluster
0/1/2
1/1/2
2/1/2
NS0
0/1/1
NS1
1/1/1
NS2
2/1/1
0/1/3
1/1/3
2/1/3
Switch (SW1)
Cluster backplane
Server data plane
Switch (SW1)
Server

- NS0 - nodeId: 0, NSIP: 10.102.29.60

- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with a common switch for the client, server, and backplane

1. Create a cluster of nodes NS0, NS1, and NS2.
2. Log on to the first node that you want to add to the cluster and do the following:

```
create cluster instance 1
add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
enable cluster instance 1
add ns ip 10.102.29.61 255.255.255.255 -type CLIP
save ns config
reboot -warm
<!--NeedCopy-->
```

3. Log on to the cluster IP address and do the following:

```
add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
<!--NeedCopy-->
```

4. Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

```
join cluster -clip 10.102.29.61 -password nsroot
save ns config
reboot -warm
<!--NeedCopy-->
```

As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

1. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

```
//For the backplane interfaces
add vlan 10
bind vlan 10 0/1/1 1/1/1 2/1/1

//For the client-side interfaces
add vlan 20
bind vlan 20 0/1/2 1/1/2 2/1/2

//For the server-side interfaces
add vlan 30
bind vlan 30 0/1/3 1/1/3 2/1/3
<!--NeedCopy-->
```

2. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco® Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

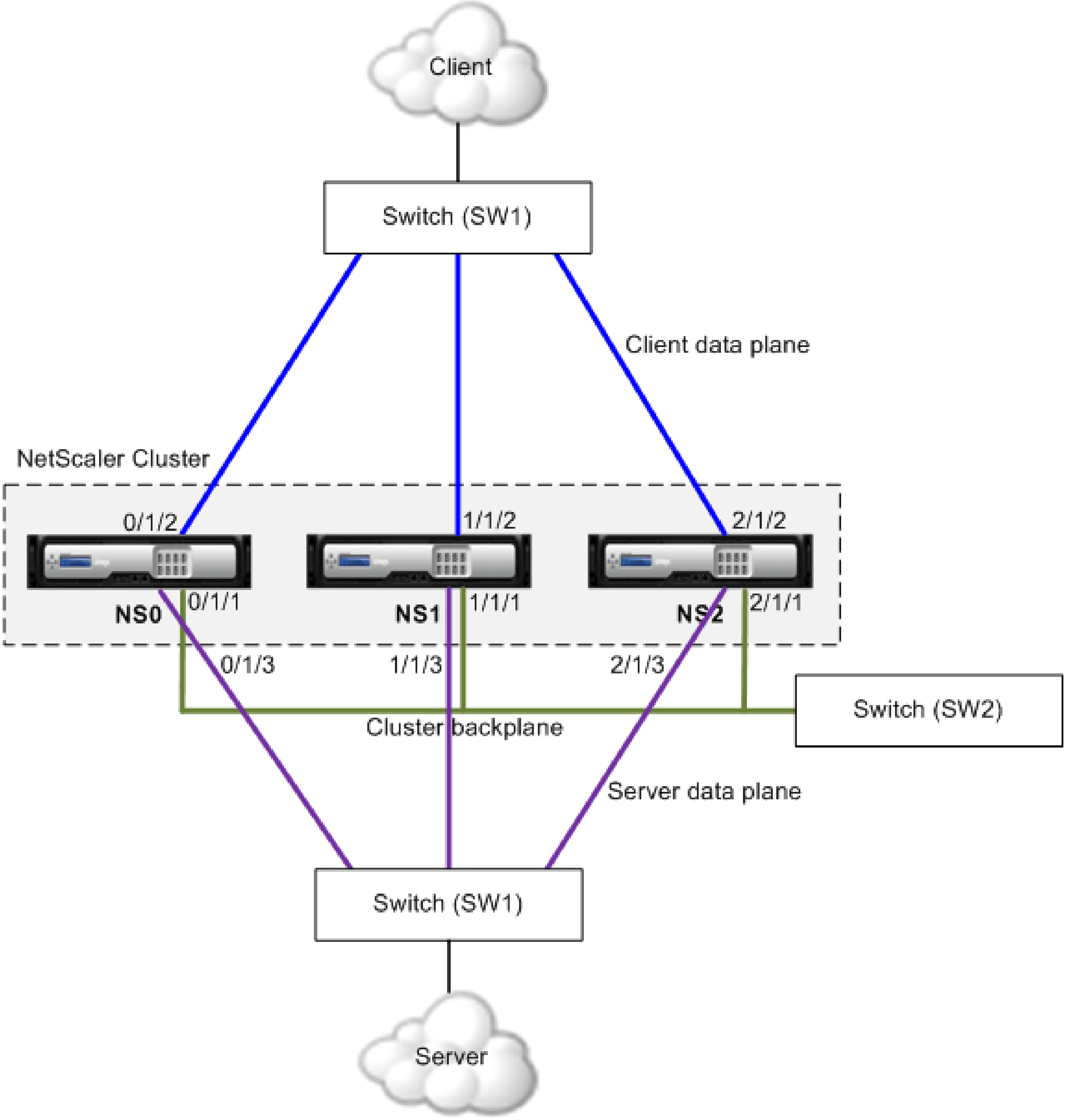
//For the client interfaces. Repeat for each interface...
interface Ethernet2/48
switchport access vlan 200
switchport mode access
end

//For the server interfaces. Repeat for each interface...
interface Ethernet2/49
switchport access vlan 300
switchport mode access
end
<!--NeedCopy-->
```

Common switch for client and server and dedicated switch for backplane

September 21, 2020

In this deployment, the clients and servers use different interfaces on the same switch to communicate with the NetScaler cluster. The cluster backplane uses a dedicated switch for inter-node communication.


Client
Switch (SW1)
Client data plane
NetScaler Cluster
0/1/2
1/1/2
2/1/2
NS0
0/1/1
NS1
1/1/1
NS2
2/1/1
0/1/3
1/1/3
2/1/3
Switch (SW2)
Cluster backplane
Server data plane
Switch (SW1)
Server

- NS0 - nodeId: 0, NSIP: 10.102.29.60
- NS1 - nodeId: 1, NSIP: 10.102.29.70
- NS2 - nodeId: 2, NSIP: 10.102.29.80

To deploy a cluster with the same switch for the clients and servers and a different switch for the cluster backplane

1. Create a cluster of nodes NS0, NS1, and NS2.

   - Log on to the first node that you want to add to the cluster and do the following:

     ```
     create cluster instance 1
     add cluster node 0 10.102.29.60 -state ACTIVE -backplane 0/1/1
     enable cluster instance 1
     add ns ip 10.102.29.61 255.255.255.255 -type CLIP
     save ns config
     reboot -warm
     <!--NeedCopy-->
     ```

   - Log on to the cluster IP address and do the following:

     ```
     add cluster node 1 10.102.29.70 -state ACTIVE -backplane 1/1/1
     add cluster node 2 10.102.29.80 -state ACTIVE -backplane 2/1/1
     <!--NeedCopy-->
     ```

   - Log on to the nodes 10.102.29.70 and 10.102.29.80 to join the nodes to the cluster.

     ```
     join cluster -clip 10.102.29.61 -password nsroot
     save ns config
     reboot -warm
     <!--NeedCopy-->
     ```

   As seen in the above commands the interfaces 0/1/1, 1/1/1, and 2/1/1 are configured as the backplane interfaces of the three cluster nodes.

2. On the cluster IP address, create VLANs for the backplane, client, and server interfaces.

   ```
   //For the backplane interfaces
   add vlan 10
   bind vlan 10 0/1/1 1/1/1 2/1/1

   //For the client-side interfaces
   add vlan 20
   bind vlan 20 0/1/2 1/1/2 2/1/2

   //For the server-side interfaces
   add vlan 30
   bind vlan 30 0/1/3 1/1/3 2/1/3
   <!--NeedCopy-->
   ```

3. On the switch, create VLANs for the interfaces corresponding to the backplane interfaces and the client and server interfaces. The following sample configurations are provided for the Cisco®

Nexus 7000 C7010 Release 5.2(1) switch. Similar configurations must be performed on other switches.

```
//For the backplane interfaces. Repeat for each interface...
interface Ethernet2/47
switchport access vlan 100
switchport mode access
end

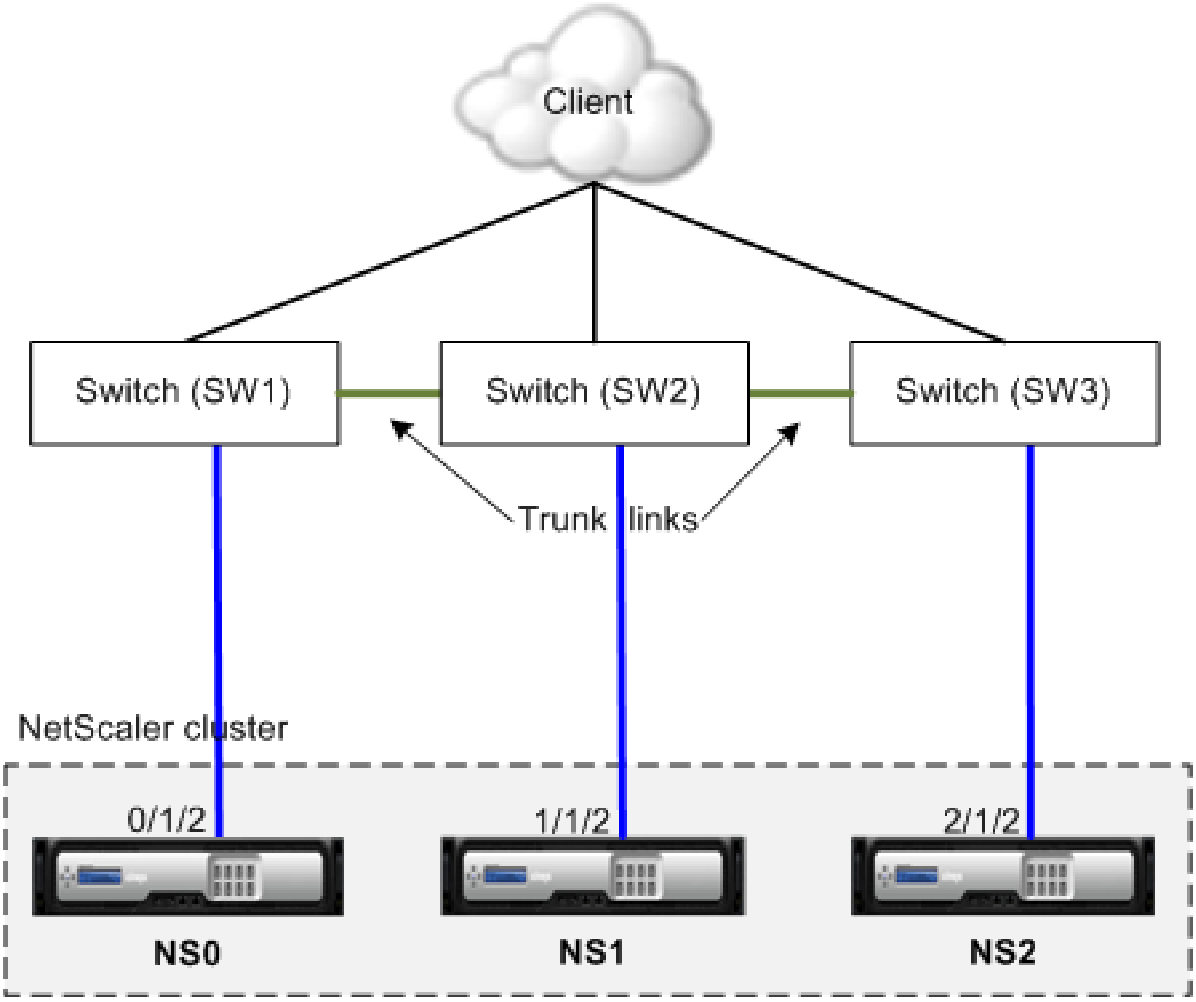
//For the client interfaces. Repeat for each interface...
interface Ethernet2/48
switchport access vlan 200
switchport mode access
end

//For the server interfaces. Repeat for each interface...
interface Ethernet2/49
switchport access vlan 300
switchport mode access
end
<!--NeedCopy-->
```

Different switch for every node

September 21, 2020

In this deployment, each cluster node is connected to a different switch and trunk links are configured between the switches.


Client
Switch (SW1)
Switch (SW2)
Switch (SW3)
Trunk links
NetScaler cluster
0/1/2
1/1/2
2/1/2
NS0
NS1
NS2

The cluster configurations will be the same as the other deployments scenarios. Most of the client-side configurations will be done on the client-side switches.

Sample cluster configurations

September 21, 2020

The following example can be used to configure a four-node cluster with ECMP
, cluster LA, or Linksets.

1. Create the cluster.

    - Log on to first node.
    - Add the cluster instance.

    ```
    add cluster instance 1
    ```

```
<!--NeedCopy-->
```

- Add the first node to the cluster.

```
add cluster node 0 10.102.33.184 -backplane 0/1/1
<!--NeedCopy-->
```

- Enable the cluster instance.

```
enable cluster instance 1
<!--NeedCopy-->
```

- Add the cluster IP address.

```
add ns ip 10.102.33.185 255.255.255.255 -type CLIP
<!--NeedCopy-->
```

- Save the configurations.

```
save ns config
<!--NeedCopy-->
```

- Warm reboot the appliance.

```
reboot -warm
<!--NeedCopy-->
```

2. Add the other three nodes to the cluster.

   - Log on to cluster IP address.
   - Add the second node to the cluster.

   ```
   add cluster node 1 10.102.33.187 -backplane 1/1/1
   <!--NeedCopy-->
   ```

   - Add the third node to the cluster.

   ```
   add cluster node 2 10.102.33.188 -backplane 2/1/1
   <!--NeedCopy-->
   ```

   - Add the fourth node to the cluster.

   ```
   add cluster node 3 10.102.33.189 -backplane 3/1/1
   <!--NeedCopy-->
   ```

3. Join the added nodes to the cluster. This step is not applicable for the first node.

    - Log on to each newly added node.
    - Join the node to the cluster.

    ```
    join cluster -clip 10.102.33.185 -password nsroot
    <!--NeedCopy-->
    ```

    - Save the configuration.

    ```
    save ns config
    <!--NeedCopy-->
    ```

    - Warm reboot the appliance.

    ```
    reboot -warm
    <!--NeedCopy-->
    ```

4. Configure the NetScaler cluster through the cluster IP address.

    ```
    // Enable load balancing feature
    enable ns feature lb

    // Add a load balancing virtual server
    add lb vserver first_lbvserver http
    ....
    ....
    <!--NeedCopy-->
    ```

5. Configure any one of the following (ECMP, cluster LA, or Linkset) traffic distribution mechanisms for the cluster.

    ECMP

    - Log on to the cluster IP address.
    - Enable the OSPF routing protocol.

    ```
        enable ns feature ospf
    <!--NeedCopy-->
    ```

    - Add a VLAN.

    ```
        add vlan 97
    <!--NeedCopy-->
    ```

    - Bind the interfaces of the cluster nodes to the VLAN.

```
    bind vlan 97 -ifnum 0/1/4 1/1/4 2/1/4 3/1/4
<!--NeedCopy-->
```

- Add a spotted SNIP on each node and enable dynamic routing on it.

```
    add ns ip 1.1.1.10 255.255.255.0 -ownerNode 0 -dynamicRouting ENABLED
    add ns ip 1.1.1.11 255.255.255.0 -ownerNode 1 -dynamicRouting ENABLED
    add ns ip 1.1.1.12 255.255.255.0 -ownerNode 2 -dynamicRouting ENABLED
    add ns ip 1.1.1.13 255.255.255.0 -ownerNode 3 -dynamicRouting ENABLED
<!--NeedCopy-->
```

- Bind one of the SNIP addresses to the VLAN.

```
    bind vlan 97 -ipAddress 1.1.1.10 255.255.255.0
<!--NeedCopy-->
```

- Configure the routing protocol on ZebOS by using vtysh shell.

  Static cluster LA

  - Log on to the cluster IP address.
  - Add a cluster LA channel.

```
    add channel CLA/1 -speed 1000
<!--NeedCopy-->
```

  - Bind the interfaces to the cluster LA channel.

```
    bind channel CLA/1 0/1/5 1/1/5 2/1/5 3/1/5
<!--NeedCopy-->
```

  - Perform equivalent configuration on the switch.

    Dynamic cluster LA

    * Log on to the cluster IP address.
    * Add the interfaces to the cluster LA channel.

```
set interface 0/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
set interface 1/1/5 -lacpmode active -lacpkey 5 -lagtype cluster
set interface 2/1/5 -lacpmode active -lacpkey 5 -lagtype
   cluster
set interface 3/1/5 -lacpmode active -lacpkey 5 -lagtype
   cluster
<!--NeedCopy-->
```

- Perform equivalent configuration on the switch.

Linksets. Assume that the node with nodeId 3 is not connected to the switch. You must configure a linkset so that the unconnected node can use the other node interfaces to communicate with the switch.

a) Log on to the cluster IP address.

b) Add a linkset.

```
add linkset LS/1
<!--NeedCopy-->
```

c) Bind the connected interfaces to the linkset.

```
bind linkset LS/1 -ifnum 0/1/6 1/1/6 2/1/6
<!--NeedCopy-->
```

6. Update the state of the cluster nodes to ACTIVE.

```
set cluster node 0 -state ACTIVE
set cluster node 1 -state ACTIVE
set cluster node 2 -state ACTIVE
set cluster node 3 -state ACTIVE
<!--NeedCopy-->
```

Monitoring services in a cluster using path monitoring

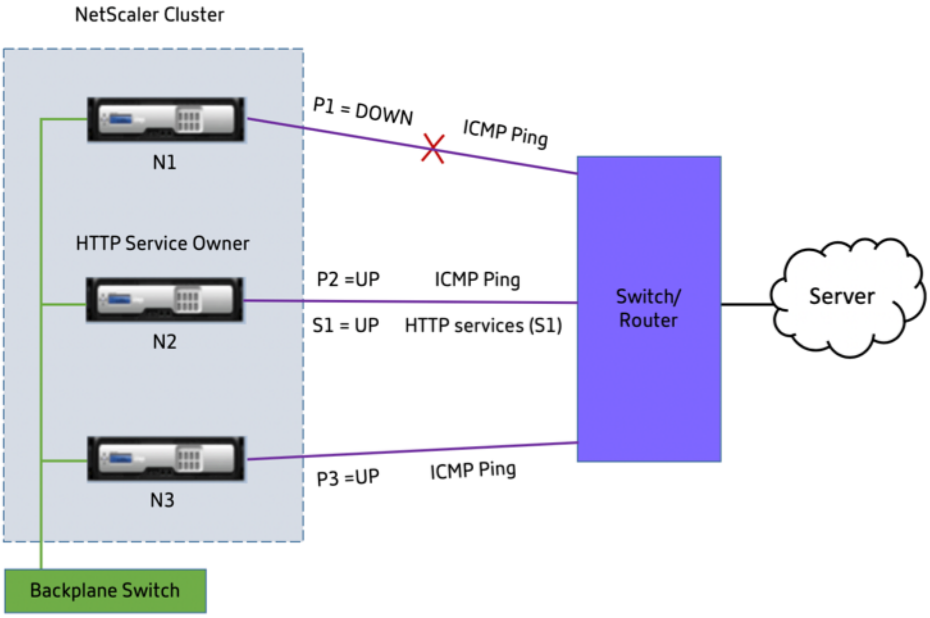
September 21, 2020

In a cluster setup, ownership for monitoring services is distributed among the nodes. Therefore, different nodes monitor different services. The node that monitors a service is called the service owner. Only the service owner probes the server to monitor the status of the services assigned to it. It further communicates the status of the services to all other nodes within the cluster. The drawback with distributed monitoring is that the network connectivity and link state between all nodes and the server is not determined. To overcome this drawback, you can use path monitoring.

Path monitoring checks the network connectivity and link state between a node and service provided by the server. A node sends ICMP pings to verify whether the server is reachable or not.

How path monitoring works

Consider an example of a NetScaler cluster consisting of three nodes N1, N2, and N3. N2 is the service owner that monitors the state of HTTP services (S1). It advertises the service state to other nodes in the cluster. Path monitoring is enabled on all nodes in the cluster, for all the services. Each node sends only an ICMP ping to the server. The service owner sends both the HTTP service request and an ICMP ping. Each node reports its path monitoring state to the service owner.


NetScaler Cluster
P1 = DOWN
ICMP Ping
N1
HTTP Service Owner
P2 =UP
ICMP Ping
S1 = UP
HTTP services (S1)
N2
Switch/ Router
Server
P3 =UP
ICMP Ping
N3
Backplane Switch

The following two parameters determine the service state of a node:

- S = service state advertised by the service owner
- P = path monitoring state of each node

Whether a node can reach a server or not, determines the path monitoring state for that node.

The following table shows the service state set based on the path monitoring state, when pathMonitorIndv parameter is enabled or disabled.

Parameter	Path monitoring state	Service state
pathMonitorIndv = NO; This is the default configuration.	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = DOWN

Parameter	Path monitoring state	Service state
pathMonitorIndv = YES	P3 = UP	S1 = DOWN
	P1 = DOWN	S1 = DOWN
	P2 = UP	S1 = UP
	P3 = UP	S1 = UP

In this example, the service owner decides the service state for all the nodes based on the node whose path monitoring state is set to DOWN. If the path monitoring state for any of the nodes is DOWN, then the service owner sets the service state for all the nodes as DOWN. The service state for all the nodes is set to UP only if the path monitoring state for each of the node is UP.

You can use path monitoring for individual nodes by enabling the pathMonitorIndv parameter. This parameter enables the service owner to set the service state for each node based on the path monitoring state of that respective node.

> **Note**
>
> If pathMonitorIndv parameter is set, some features like persistence might break.

Configuring path monitoring

Path monitoring is applicable for all the services and service groups. Path monitoring parameter is disabled by default.

To enable path monitoring for services/service groups by using the CLI

At the command prompt, type:

```
add service <service name> <IP address> <service type> <port> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]

add servicegroup <servicegroup name> <service type> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

Example:

```
add service s1 1.1.1.1 HTTP 80 -pathMonitor YES
add servicegroup sg_1 HTTP -pathMonitor YES

add service s1 1.1.1.1 HTTP 80 -pathMonitor YES -pathMonitorIndv YES
add servicegroup sg_1 HTTP -pathMonitor YES -pathMonitorIndv YES
<!--NeedCopy-->
```

You can also set the path monitoring parameter from the set command, as follows:

```
set service <service name> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]
set servicegroup <servicegroup name> [-pathMonitor <YES | NO>] [-pathMonitorIndv <YES | NO>]
<!--NeedCopy-->
```

Example:

```
set service s1 -pathMonitor YES
set servicegroup sg_1 -pathMonitor YES


set service s1 -pathMonitorIndv YES
set servicegroup sg_1 -pathMonitorIndv NO
<!--NeedCopy-->
```

To enable path monitoring for services/service groups by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.

   For service groups, navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.

2. In the **Services**/**Service Groups** pane, select a service/service group from the list, and then double-click to open it.
3. On the **Service Settings** tab, click **Edit**.
4. Select **Path Monitoring**.
5. Select **Individual Path Monitoring**, if you want to apply it, and then click **OK**.

> **Note**
>
> You can enable Individual Path Monitoring only if you enable Path Monitoring.

Upgrading or downgrading the NetScaler cluster

September 21, 2020

All the nodes of a NetScaler cluster must be running the same software version. Therefore, to upgrade or downgrade the cluster, you must upgrade or downgrade each NetScaler appliance of the cluster, one node at a time.

A node that is being upgraded or downgraded is not removed from the cluster. The node continues to be a part of the cluster and serves traffic uninterrupted, except for the down-time when the node reboots after it is upgraded or downgraded.

However, due to software version mismatch among the cluster nodes, configuration propagation is disabled on the cluster and is enabled only after all the cluster nodes are of the same version. Since configuration propagation is disabled during upgrading on downgrading a cluster, you cannot perform any configurations through the cluster IP address during this time.

> **Important**
>
> - Configurations can be lost on downgrading the cluster.
> - When you are upgrading a cluster to NetScaler 11.0 build 64.x from an earlier NetScaler 11.0 build, cluster configuration propagation is disabled. This exception arises because the cluster version in build 64.x is different from the one in previous NetScaler 11.0 builds.
>
> Traditionally, this issue occurred only during an upgrade of a cluster to a different NetScaler version (for example, from 10.5 to 11.0). It must be noted that normally the cluster version matches the NetScaler version.
>
> **Note**
>
> Configuration propagation remains disabled until all the cluster nodes are upgraded to Build 64.x.

Points to note before upgrading or downgrading the cluster

- You cannot add cluster nodes while upgrading or downgrading the cluster software version.
- You can perform node-level configurations through the NSIP address of individual nodes, but make sure to perform the same configurations on all the nodes to maintain them in synch.
- You cannot execute the "start nstrace" command from the cluster IP address when the cluster is being upgraded. However, you can get the trace of individual nodes by performing this operation on individual cluster nodes through their NSIP address.
- Owing to changes in cluster licensing that were made in NetScaler 10.5 Build 52.11 see Prerequisites for cluster nodes, look into the following:
  - If the cluster is setup in a build prior to NetScaler 10.5 Build 52.11, the cluster will work with the separate cluster license file. No changes are required.

  - If the cluster is setup in NetScaler 10.5 Build 52.11 or later releases and then downgraded to a build prior to NetScaler 10.5 Build 52.11, the downgraded cluster will not work as it now expects a separate cluster license file.

- While upgrading from any NetScaler 10.1 build to a later release, syncookie must be disabled on all TCP profiles (using the "set ns tcpProfile <name> -synCookie DISABLED" command) and after that a striped SNIP must be added on the CLIP subnet. Once upgraded, syncookie can be enabled again.

- While upgrading the NetScaler appliance from a NetScaler 10.1 build to a NetScaler 10.5 build, do not execute the "show audit messages" command as this can cause the NetScaler appliance to become unresponsive.

- NetScaler 10.5 54.x and 55.x builds are not suitable for cluster deployment. This is because, for services that need probing, SYN packets are processed locally (on the flow receiver) even though syncookie is disabled.

- When a cluster is being upgraded, it is possible that the upgraded nodes have some additional features activated that are not available on the nodes that are not upgraded. This results in a license mismatch warning while the cluster is being upgraded. This warning will be automatically resolved when all the cluster nodes are upgraded.

> **Important**
>
> - Citrix recommends that you wait for the previous node to become active before upgrading or downgrading the next node.
>
> - Citrix recommends that the cluster configuration node must be upgraded/downgraded last to avoid multiple disconnect of cluster IP sessions.

To upgrade or downgrade the software of the cluster nodes

1. Make sure the cluster is stable and the configurations are synchronized on all the nodes.

2. Access each node through its NSIP address and perform the following:

   - Upgrade or downgrade the cluster node. For detailed information about upgrading and downgrading the software of an appliance, see Upgrade and downgrade a NetScaler appliance.

   - Save the configurations.

   - Reboot the appliance.

3. Repeat step 2 for each of the other cluster nodes.

Operations supported on individual cluster nodes

September 21, 2020

As a rule, NetScaler appliances that are a part of a cluster cannot be individually configured from their NSIP address. However, there are some operations that are an exception to this rule. These operations, when executed from the NSIP address, are not propagated to other cluster nodes.

The operations are:

cluster instance (set	rm	enable	disable)

-

cluster node (set	rm)

-

nstrace (start	show	stop)

-

interface (set	enable	disable)

-

route (add	rm	set	unset)

-

arp (add	rm	send -all)

-
- force cluster sync
- sync cluster files

- disable ntp sync
- save ns config
- reboot
- shutdown

For example, when you execute the command disable interface 1/1/1 from the NSIP address of a cluster node, the interface is disabled only on that node. Since the command is not propagated, the interface 1/1/1 remains enabled on all the other cluster nodes.

FAQs

September 21, 2020

A list of the frequently asked questions about clustering (NetScaler 11.0, 10.5, 10.1). Click a question to view its answer.

How many NetScaler appliances can be included in a single NetScaler cluster?

A NetScaler cluster can include one appliance or as many as 32 NetScaler appliance nCore hardware or virtual appliances. Each of these nodes must satisfy the criteria specified in Prerequisites for Cluster Nodes.

Can a NetScaler appliance be a part of multiple clusters?

No. A NetScaler appliance can belong to can belong to one cluster only.

What is a cluster IP address? What is its subnet mask?

The cluster IP address is the management address of a NetScaler cluster. All cluster configurations must be performed by accessing the cluster through this address. The subnet mask of the cluster IP address is fixed at 255.255.255.255.

How can I make a specific cluster node as the cluster configuration coordinator?

To manually set a specific node as the cluster configuration coordinator, you must set the priority of that node to the lowest numeric value (highest priority). To understand, let us consider a cluster with three nodes that have the following priorities:

n1 - 29, n2 - 30, n3 - 31

Here, n1 is the configuration coordinator. If you want to make n2 the configuration coordinator, you must set its priority to a value that is lower than n1, for example, 28. On saving the configuration, n2 becomes the configuration coordinator.

> **Note**
>
> n2 with its original priority value of 30 will also become the configuration coordinator when n1 goes down, as the node with the next lowest priority value is selected in the event that the configuration coordinator goes down.

Why are the network interfaces of a cluster represented in 3-tuple (n/u/c) notation instead of the regular 2-tuple (u/c) notation?

When a NetScaler appliance is part of a cluster, you must be able to identify the node to which the interface belongs. Therefore, the network interface naming convention for cluster nodes is modified from u/c to n/u/c, where n denotes the node Id.

How can I set the hostname for a cluster node?

The hostname of a cluster node must be specified by executing the **set ns hostname** command through the cluster IP address. For example, to set the hostname of the cluster node with ID 2, the command is:

set ns hostname hostName1 -**ownerNode** 2

Can I automatically detect NetScaler appliances so that I can add them to a cluster?

Yes. The configuration utility allows you to discover appliances that are present in the same subnet as the NSIP address of the configuration coordinator. For more information, see Discovering NetScaler Appliances.

Is the traffic serving capability of a cluster affected if a node is removed or disabled or reboot or shutdown or made inactive?

Yes. When any of these operations are performed on an active node of the cluster, the cluster will have one less node to serve traffic. Also, existing connections on this node are terminated.

I have multiple standalone appliances, each of which has different configurations. Can I add them to a single cluster?

Yes. You can add appliances that have different configurations to a single cluster. However, when the appliance is added to the cluster, the existing configurations are cleared. To use the configurations that are available on each of the individual appliances, you must:

1. Create a single *.conf file for all the configurations.
2. Edit the configuration file to remove features that are not supported in a cluster environment.
3. Update the naming convention of interfaces from 2-tuple (u/c) format to 3-tuple (n/u/c) format.
4. Apply the configurations to the configuration coordinator node of the cluster by using the batch-command.

Can I migrate the configurations of a standalone NetScaler appliance or an HA setup to the clustered setup?

No. When a node is added to a clustered setup, its configurations are implicitly cleared by using the **clear ns config** command (with the **extended** option). In addition, the SNIP addresses and all VLAN configurations (except default VLAN and NSVLAN) are cleared. Therefore, it is recommended that you back up the configurations before adding the appliance to a cluster. Before using the backed-up configuration file for the cluster, you must:

1. Edit the configuration file to remove features that are not supported in a cluster environment.
2. Update the naming convention of interfaces from two-tuple (x/y) format to three-tuple (x/y/z) format.
3. Apply the configurations to the configuration coordinator node of the cluster by using the **batch** command.

Are backplane interfaces part of the L3 VLANs?

Yes, by default, backplane interfaces have presence on all the L3 VLANs that are configured on the cluster.

How can I configure a cluster that includes nodes from different networks?

> **Note**
>
> Supported from NetScaler 11.0 onwards.

A cluster that includes nodes from different networks is called a L3 cluster (sometimes referred to as a cluster in INC mode). In an L3 cluster, all nodes that belong to a single network must be grouped together in a single nodegroup. Therefore, if a cluster includes two nodes each from three different networks, you will have to create 3 nodegroups (one for each network) and associate each of these nodegroups with the nodes that belong to that network. For configuration information, see the steps to setup a cluster.

How can I configure/un-configure the NSVLAN on a cluster?

Do either one of the following:

- To make the NSVLAN available in a cluster, make sure that each appliance has the same NSVLAN configured before it is added to cluster.
- To remove the NSVLAN from a cluster node, first remove the node from the cluster and then delete the NSVLAN from the appliance.

I have a cluster setup where some NetScaler appliance nodes are not connected to the external network. Can the cluster still function normally?

Yes. The cluster supports a mechanism called linksets, which allows unconnected nodes to serve traffic by using the interfaces of connected nodes. The unconnected nodes communicate with the connected nodes through the cluster backplane. For more information, see Using Linksets.

How can deployments that require MAC-Based Forwarding (MBF) be supported in a clustered setup?

Deployments that use MBF must use linksets. For more information, see Using Linksets.

Can I execute commands from the NSIP address of a cluster node?

No. Access to individual cluster nodes through the NSIP addresses is read-only. Therefore, when you log on to the NSIP address of a cluster node you can only view the configurations and the statistics. You cannot configure anything. However, there are some operations you can execute from the NSIP address of a cluster node. For more information, see Operations Supported on Individual Nodes.

Can I disable configuration propagation among cluster nodes?

No, you cannot explicitly disable the propagation of cluster configurations among cluster nodes. However, during a software upgrade or downgrade, a version mismatch error can automatically disable configuration propagation.

Can I change the NSIP address or change the NSVLAN of a NetScaler appliance when it is a part of the cluster?

No. To make such changes you must first remove the appliance from the cluster, perform the changes, and then add the appliance to the cluster.

Does the NetScaler cluster support L2 and L3 VLANs?

Yes. A cluster supports VLANs between cluster nodes. The VLANs must be configured on the cluster IP address.

- **L2 VLAN.** You can create a layer2 VLAN by binding interfaces that belong to different nodes of the cluster.
- **L3 VLAN.** You can create a layer3 VLAN by binding IP addresses that belong to different nodes of the cluster. The IP addresses must belong to the same subnet. Make sure that one of the following criteria is satisfied. Otherwise, the L3 VLAN bindings can fail.
  - All nodes have an IP address on the same subnet as the one bound to the VLAN.
  - The cluster has a striped IP address and the subnet of that IP address is bound to the VLAN.

When you add a new node to a cluster that has only spotted IPs, the sync happens before spotted IP addresses are assigned to that node. In such cases, L3 VLAN bindings can be lost. To avoid this loss, either add a striped IP or add the L3 VLAN bindings on the NSIP of the newly added node.

How can I configure SNMP on a NetScaler cluster?

SNMP monitors the cluster, and all the nodes of the cluster, in the same way that it monitors a standalone appliance. The only difference is that SNMP on a cluster must be configured through the cluster IP address. When generating hardware specific traps, two additional varbinds are included to identify the node of the cluster: node ID and NSIP address of the node.

What details must I have available when I contact technical support for cluster-related issues?

The NetScaler appliance provides a **show techsupport -scope cluster** command that extracts configuration data, statistical information, and logs of all the cluster nodes. You must run this command on the cluster IP address.

The output of this command is saved in a file named *collector_cluster_<nsip_CCO>_P_<date-timestamp>.tar.gz* which is available in the */var/tmp/support/cluster/* directory of the configuration coordinator.

Send this archive to the technical support team to debug the issue.

Can I use striped IP addresses as the default gateway of servers?

In case of cluster deployments, make sure the default gateway of the server points to a striped IP address (if you are using a NetScaler appliance-owned IP address). For example, in case of LB deployments with USIP enabled, the default gateway must be a striped SNIP address.

Can I view routing configurations of a specific cluster node from the cluster IP address?

Yes. You can view and clear the configurations specific to a node by specifying the owner node while entering the vtysh shell.

For example, to view the output of a command on nodes 0 and 1, the command is as follows:

```
\> vtysh

ns# owner-node 0 1
ns(node-0 1)\# show cluster state
ns(node-0 1)\# exit-cluster-node
ns\#
<!--NeedCopy-->
```

How can I specify the node for which I want to set the LACP system priority?

> **Note**
>
> Supported from NetScaler 10.1 onwards.

In a cluster, you must set that node as the owner node by using the **set lacp** command.

For example: To set the LACP system priority for node with ID 2:

"'set lacp -sysPriority 5 -ownerNode 2

```
## How are IP tunnels configured in a cluster setup?

> **Note**
>
> Supported from NetScaler 10.1 onwards.

Configuring IP tunnels in a cluster is the same as on a standalone
    appliance. The only difference is that in a cluster setup, the local
     IP address must be a striped SNIP address.

## How can I add a failover interface set (FIS) on the nodes of a
    NetScaler cluster?

> **Note**
>
> Supported from NetScaler 10.5 onwards.

On the cluster IP address, specify the ID of the cluster node on which
    the FIS must be added, using the command as follows:

```

~~~
```add fis <name> -ownerNode <nodeId><!--NeedCopy-->
~~~

> **Notes**
>
> - The FIS name for each cluster node must be unique.
> - A cluster LA channel can be added to a FIS. You must make sure that the cluster LA channel has a local interface as a member interface.

For more information on FIS, see Configuring failover interface set.

How are net profiles configured in a cluster setup?

> **Note**
>
> Supported from NetScaler 10.5 onwards.

You can bind spotted IP addresses to a net profile. This net profile can then be bound to spotted load balancing virtual server or service (that is defined using a nodegroup). The following recommendations must be followed, failing which, the net profile configurations are not honored and the USIP/USNIP settings will be used:

> **Note**
>
> - If the **strict** parameter of the nodegroup is set to **Yes**, the net profile must contain a minimum of one IP address from each nodegroup member.
> - If the **strict** parameter of the nodegroup is set to **No**, the net profile must include at least one IP address from each of the cluster nodes.

How can I configure WIonNS in a cluster setup?

> **Note**
>
> Supported from NetScaler 11.0 Build 62.x onwards.

To use WIonNS on a cluster, you must do the following:

1. Make sure that the Java package and the WI package are present in the same directory on all the cluster nodes.
2. Create a load balancing virtual server that has persistency configured.
3. Create services with IP addresses as the NSIP address of each of the cluster nodes that you want to serve WI traffic. This step can only be configured using the NetScaler CLI.
4. Bind the services to the load balancing virtual server.

> **Note**
>
> If you are using WIonNS over a VPN connection, make sure that the load balancing virtual server is set as WIHOME.

Can the cluster LA channel be used for management access?

No. Management access to a cluster node, must not be configured on a cluster LA channel (for example, CLA/1) or its member interfaces. This is because when the node is INACTIVE, the corresponding cluster LA interface is marked as power down and therefore looses management access.

How cluster nodes communicate to each other and what are the different types of traffic that goes through backplane?

A backplane is a set of interfaces in which one interface of each node is connected to a common switch, which is called the cluster backplane switch. The different types of traffic that goes through backplane, which is used by internode communication are:

- Node to Node Messaging (NNM)
- Steered traffic
- Configuration propagation and synchronization

Each node of the cluster uses a special MAC cluster backplane switch address to communicate with other nodes through the backplane. The cluster special MAC is of the form: **0x02 0x00 0x6F <cluster_id> <node_id> <reserved>**, where <cluster_id> is the cluster instance ID, <node_id> is the node number of the NetScaler appliance that are added to a cluster.

> **Note**
>
> The amount of traffic that is handled by a backplane will have negligible CPU overhead.

What gets routed over GRE tunnel for Layer 3 cluster?

Only steered data traffic goes over the GRE tunnel. The packets are steered through the GRE tunnel to the node on the other subnet.

How Node to Node Messaging (NNM) and heartbeat messages are exchanged, and how are they routed?

The NNM, heartbeat messages, and cluster protocol are non-steering traffic. These messages are not sent through the tunnel, but they are routed directly.

What are the MTU recommendations when Jumbo frames are enabled for layer 3 cluster tunneled traffic?

The following are the layer 3 cluster recommendations of Jumbo MTU over GRE tunnel:

- The Jumbo MTU can be configured among cluster nodes across the L3 path to accommodate GRE tunnel overhead.

- The fragmentation does not happen for full sized packets which need to be steered.
- Steering of traffic continues to work even if Jumbo frames are not allowed, but with more overhead due to fragmentation.

How the global hash key is generated and shared across all nodes?

The rsskey for a standalone appliance is generated at the boot time. In a cluster setup, the first node holds the rsskey of the cluster. Every new node joining to the cluster will synchronize the rsskey.

What is the need of "set rsskeytype –rsskey symmetric" command for *:*, USIP on, useproxyport off, topologies?

This is not specific to cluster, applies to a standalone appliance as well. With USIP ON, and use proxy port disabled, symmetric rsskey reduces both Core to Core (C2C) steering and node to node steering.

Troubleshooting the NetScaler cluster

September 21, 2020

If a failure occurs in a NetScaler cluster, the first step in troubleshooting is to get information on the cluster instance and the cluster nodes by running the show cluster instance <clId> and show cluster node <nodeId> commands respectively.

If you are not able to find the issue by using the above two approaches, you can use one of the following:

- **Isolate the source of the failure.** Try bypassing the cluster to reach the server. If the attempt is successful, the problem is probably with the cluster setup.
- **Check the commands recently executed.** Run the history command to check the recent configurations performed on the cluster. You can also review the ns.conf file to verify the configurations that have been implemented.
- **Check the ns.log files.** Use the log files, available in the /var/log/ directory of each node, to identify the commands executed, status of commands, and the state changes.
- **Check the newnslog files.** Use the newnslog files, available in the /var/nslog/ directory of each node, to identify the events that have occurred on the cluster nodes. You can view multiple newnslog files as a single file, by copying the files to a single directory, and then running the following command:

  ```
  nsconmsg -K newnslog-node<id> -K newnslog.node<id> -d current
  <!--NeedCopy-->
  ```

If you still cannot resolve the issue, you can try tracing the packets on the cluster or use the show techsupport -scope cluster command to send the report to the technical support team.

Tracing the packets of a NetScaler cluster

September 21, 2020

The NetScaler appliance operating system provides a utility called *nstrace* to get a dump of the packets that are received and sent out by an appliance. The utility stores the packets in trace files. You can use these files to debug problems in the flow of packets to the cluster nodes. The trace files must be viewed with the Wireshark application.

Some salient aspects of the nstrace utility are:

- Can be configured to trace packets selectively by using classic expressions and default expressions.
- Can capture the trace in multiple formats: nstrace format (.cap) and TCP dump format (.pcap).
- Can aggregate the trace files of all cluster nodes on the configuration coordinator.
- Can merge multiple trace files into a single trace file (only for .cap files).

You can use the nstrace utility from the NetScaler appliance command line or the NetScaler appliance shell.

To trace packets of a standalone appliance

Run the start nstrace command on the appliance. The command creates trace files in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id>.cap.

You can view the status by executing the show nstrace command. You can stop tracing the packets by executing the stop nstrace command.

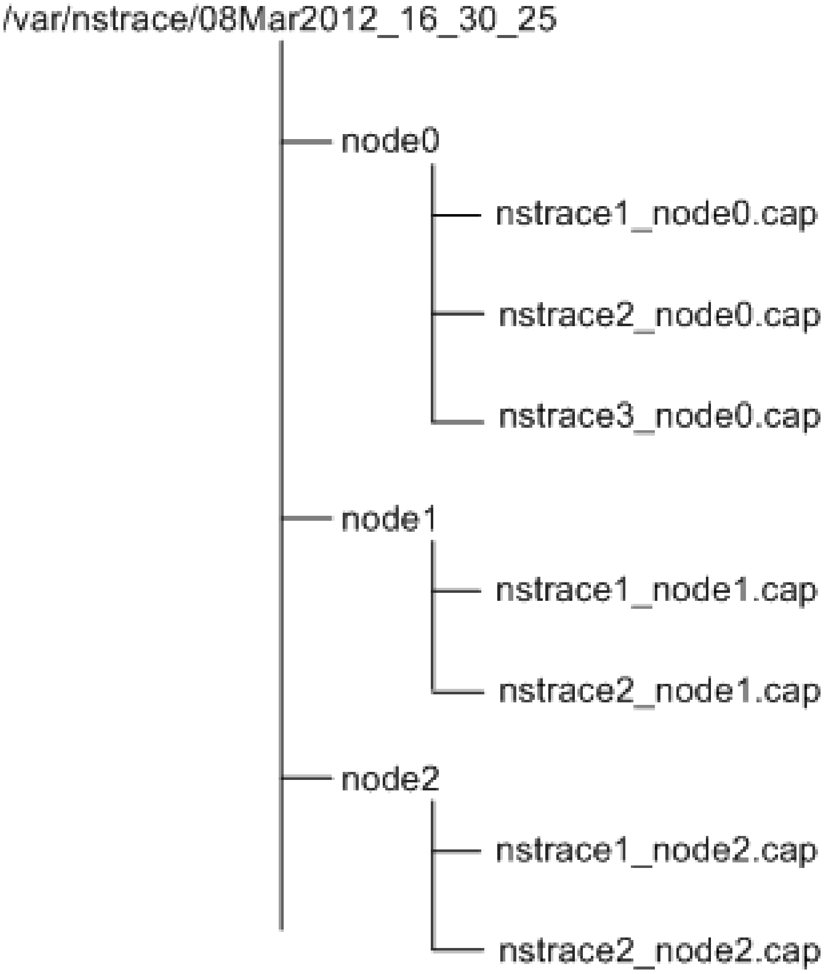
Note: You can also run the nstrace utility from the NetScaler appliance shell by executing the nstrace.sh file. However, it is recommended that you use the nstrace utility through the NetScaler appliance command line interface.

To trace packets of a cluster

You can trace the packets on all the cluster nodes and obtain all the trace files on the configuration coordinator.

Run the start nstrace command on the cluster IP address. The command is propagated and executed on all the cluster nodes. The trace files are stored in individual cluster nodes in the /var/nstrace/<date-timestamp> directory. The trace file names are of the form nstrace<id>_node<id>.cap.

You can use the trace files of each node to debug the nodes operations. But if you want the trace files of all cluster nodes in one location, you must run the stop nstrace command on the cluster IP address. The trace files of all the nodes are downloaded on the cluster configuration coordinator in the /var/nstrace/<date-timestamp> directory as follows:


/var/nstrace/08Mar2012_16_30_25
node0
nstrace1_node0.cap
nstrace2_node0.cap
nstrace3_node0.cap
node1
nstrace1_node1.cap
nstrace2_node1.cap
node2
nstrace1_node2.cap
nstrace2_node2.cap

Merge multiple trace files

You can prepare a single file from the trace files (supported only for .cap files) obtained from the cluster nodes. The single trace files gives you a cumulative view of the trace of the cluster packets. The trace entries in the single trace file are sorted based on the time the packets were received on the cluster.

To merge the trace files, at the NetScaler appliance shell, type:

nstracemerge.sh -srcdir <DIR> -dstdir <DIR> -filename <name> -filesize <num>

where,

- srcdir is the directory from which the trace files are merged. All trace files within this directory are merged into a single file.

- dstdir is the directory where the merged trace file are created.
- filename is the name of the trace file that is created.
- filesize is the size of the trace file.

Examples

Following are some examples of using the nstrace utility to filter packets.

- To trace the packets on the backplane interfaces of three nodes:

  Using classic expressions:

  ```
  start nstrace -filter "INTF == 0/1/1 && INTF == 1/1/1 && INTF == 2/1/1"
  <!--NeedCopy-->
  ```

  Using default expressions:

  ```
  start nstrace -filter "CONNECTION.INTF.EQ("0/1/1") && CONNECTION.INTF.EQ("1/1/1") && CONNECTION.INTF.EQ("2/1/1")"
  <!--NeedCopy-->
  ```

- To trace the packets from a source IP address 10.102.34.201 or from a system whose source port is greater than 80 and the service name is not "s1":

  Using classic expressions

  ```
  start nstrace -filter "SOURCEIP == 10.102.34.201 || (SVCNAME != s1 && SOURCEPORT > 80)"
  <!--NeedCopy-->
  ```

  Using default expressions

  ```
  start nstrace -filter "CONNECTION.SRCIP.EQ(10.102.34.201) || (CONNECTION.SVCNAME.NE("s1") && CONNECTION.SRCPORT.GT(80))"
  <!--NeedCopy-->
  ```

Capturing SSL Session Keys During a Trace

When you run the "start nstrace" command, you can set the new "capsslkeys" parameter to capture the SSL master keys for all SSL sessions. If you include this parameter, a file named nstrace.sslkeys is generated along with the packet trace. This file can be imported into Wireshark to decrypt the SSL traffic in the corresponding trace file.

This functionality is similar to web browsers exporting session keys that can later be imported into Wireshark for decrypting SSL traffic.

Advantages of using SSL session keys

Following are the advantages of using SSL session keys:

1. Generates smaller trace files that do not include the extra packets created by the SSLPLAIN mode of capturing.
2. Provides the ability to view plaintext [SP(1] from the trace and choose whether to share the master keys file or protect sensitive data by not sharing it.

Limitations of using SSL session keys

Following are the limitations of using SSL session keys:

1. SSL sessions cannot be decrypted if initial packets of the session are not captured.
2. SSL sessions cannot be captured if the Federal Information Processing Standard (FIPS) mode is enabled.

To capture SSL session keys by using the command line interface (CLI)

At the command prompt, type the following commands to enable or disable SSL session keys in a trace file and verify trace operation.

```
> start nstrace -capsslkeys ENABLED
> show nstrace
Example
> start nstrace -capsslkeys ENABLED
> show nstrace
      State:  RUNNING          Scope:  LOCAL            TraceLocation: "/var/nstrace/04May2016_17_51_54/..."
      Nf:  24                  Time:  3600              Size:  164              Mode:  TXB NEW_RX
      Traceformat:  NSCAP      PerNIC:  DISABLED        FileName: 04May2016_17_51_54 Link:  DISABLED
      Merge:  ONSTOP           Doruntimecleanup:  ENABLED TraceBuffers: 5000      SkipRPC:  DISABLED
      SkipLocalSSH:  DISABLED  Capsslkeys:  ENABLED     InMemoryTrace: DISABLED
 Done
```

To configure SSL session keys by using the NetScaler GUI

1. Navigate to **Configuration > System > Diagnostics > Technical Support Tools** and click **Start new Trace** to start tracing encrypted packets on an appliance.

2. On the **Start Trace** page, select the **Capture SSL Master Keys** check box.
3. Click **OK** and **Done**.

To import the SSL Master Keys into Wireshark

On the Wireshark GUI, navigate to **Edit > Preferences > Protocols > SSL > (Pre)-Master-Secret log filename** and specify the master key files obtained from the appliance.

Troubleshooting common issues

September 21, 2020

While joining a node to the cluster, I get the following message, "ERROR: Invalid interface name/number." What must I do to resolve this error?

This error occurs if you provided an invalid or incorrect backplane interface while using the add cluster nodecommand to add the node. To resolve this error, verify the interface you provided while adding the node. Make sure that you have not specified the appliance's management interface as the backplane interface, and that the <nodeId> bit of the interface is the same as the node's Id. For example, if the nodeId is 3, the backplane interface must be 3/<c>/<u>.

While joining a node to the cluster, I get the following message, "ERROR: Clustering cannot be enabled, because the local node is not a member of the cluster." What must I do to resolve this error?

This error occurs when you try to join a node without adding the node's NSIP to the cluster. To resolve this error, you must first add the node's NSIP address to the cluster by using the **add cluster node** command and then execute the **join cluster** command.

While joining a node to the cluster, I get the following message, "ERROR: Connection refused." What must I do to resolve this error?

This error can occur due to the following reasons:

- **Connectivity problems.** The node cannot connect to the cluster IP address. Try pinging the cluster IP address from the node that you are trying to join.
- **Duplicate cluster IP address.** Check to see if the cluster IP address exists on some non-cluster node. If it does, create a new cluster IP address and try re-joining the cluster.

While joining a node to the cluster, I get the following message, "ERROR: License mismatch between the configuration coordinator and the local node." What must I do to resolve this error?

The appliance that you are joining to the cluster must have the same licenses as the configuration coordinator. This error occurs when the licenses on the node you are joining do not match the licenses on the configuration coordinator. To resolve this error, run the following commands on both the nodes and compare the outputs.

From the command line:

- show ns hardware
- show ns license

From the shell:

- nsconmsg -g feature -d stats
- ls /nsconfig/license
- View the contents of the /var/log/license.log file

What must I do when the configurations of a cluster node are not in synch with the cluster configurations?

In most cases, the configurations are automatically synchronized between all the cluster nodes. However, if you feel that the configurations are not synchronized on a specific node, you must force the synchronization by executing the force cluster sync command from the node that you want to synchronize. For more information, see Synchronizing Cluster Configurations.

When configuring a cluster node, I get the following message, "ERROR: Session is read-only; connect to the cluster IP address to modify the configuration."

All configurations on a cluster must be done through the cluster IP address and the configurations are propagated to the other cluster nodes. All sessions established through the NSIP address of individual nodes are read-only.

Why does the node state show "INACTIVE" when the node health shows "UP"?

A healthy node can be in the INACTIVE state for a number of reasons. A scan of ns.log or error counters can help you determine the exact reason.

How can I resolve the health of a node when its health shows "NOT UP"?

Node health "**Not UP**" indicates that there are some issues with the node. To know the root cause, you must run the **show cluster node** command. This command displays the node properties and the reason for the node failure.

What must I do when the health of a node shows as "NOT UP" and the reason indicates that configuration commands have failed on a node?

This issue arises when some commands are not executed on the cluster nodes. In such cases, you must make sure that the configurations are synchronized using one of the following options:

- If some of the cluster nodes are in this state, you must perform the force cluster synchronization operation on those nodes. For more information, see Synchronizing Cluster Configurations.

- If all cluster nodes are in this state, you must disable and then enable the cluster instance on all the cluster nodes.

When I run the set vserver command, I get the following message, "No such resource." What must I do to resolve this issue?

The **set vserver** command is not supported in clustering. The **unset vserver, enable vserver, disable vserver,** and **rm vserver** commands are also not supported. However, the **show vserver** command is supported.

I cannot configure the cluster over a Telnet session. What must I do?

Over a telnet session, the cluster IP address can be accessed only in read-only mode. Therefore, you cannot configure a cluster over a telnet session.

I notice a significant time difference across the cluster nodes. What must I do to resolve this issue?

When PTP packets are dropped due to backplane switch or if the physical resources are over-committed in a virtual environment, the time will not get synchronized.

To synchronize the times, you must do the following on the cluster IP address:

1. Disable PTP.

   set ptp -state disable

2. Configure Network Time Protocol (NTP) for the cluster. For more information, see Setting up Clock Synchronization.

What must I do, if there is no connectivity to the cluster IP address and the NSIP address of a cluster node?

If you cannot access to the cluster IP address or the NSIP of a cluster node, you must access the appliance through the serial console. If the NSIP address is reachable, you can SSH to the cluster IP address from the shell by executing the following command at the shell prompt:

"'# ssh nsroot@

```
**What must I do to recover a cluster node that has connectivity issues?**

To recover a node that has connectivity issues:

1.  Disable the cluster instance on that node (since you cannot execute commands from the NSIP of a cluster node).

1.  Execute the commands required to recover the node.

1.  Enable the cluster instance on that node.
```

~~~

Some nodes of the cluster have two default routes. How can I remove the second default route from the cluster node?

To delete the additional default route, do the following on each node that has the extra route:

1. Disable the cluster instance.

    ```disable cluster instance <clId><!--NeedCopy-->
~~~

1. Remove the route.

   `rm route <network> <netmask> <gateway><!--NeedCopy-->`

2. Enable the cluster instance.

   `enable cluster instance <clId><!--NeedCopy-->`

**The cluster functionality gets affected when an existing cluster node comes online. What must I do to resolve this issue?**

If RPC password of node is changed from the cluster IP address when that node is out of the cluster, then, when the node comes online, there is a mismatch in rpc credentials and this could affect cluster functionality. To solve this issue, use the set ns rpcNode command to update the password on the NSIP of the node which has come online.

## Content Switching

September 21, 2020

In today's complex Web sites, you may want to present different content to different users. For example, you may want to allow users from the IP range of a customer or partner to have access to a special Web portal. You may want to present content relevant to a specific geographical area to users from that area. You may want to present content in different languages to the speakers of those languages. You may want to present content tailored to specific devices, such as smartphones, to those who use the devices. The NetScaler content switching feature enables the appliance to distribute client requests across multiple servers on the basis of specific content that you wish to present to those users.

To configure content switching, first create a basic content switching setup, and then customize it to meet your needs. This entails enabling the content switching feature, setting up load balancing for the server or servers that host each version of the content that is being switched, creating a content switching virtual server, creating policies to choose which requests are directed to which load balancing virtual server, and binding the policies to the content switching virtual server. You can then customize the setup to meet your needs by setting precedence for your policies, protecting your setup by configuring a backup virtual server, and improving the performance of your setup by redirecting requests to a cache.

## How Content Switching Works

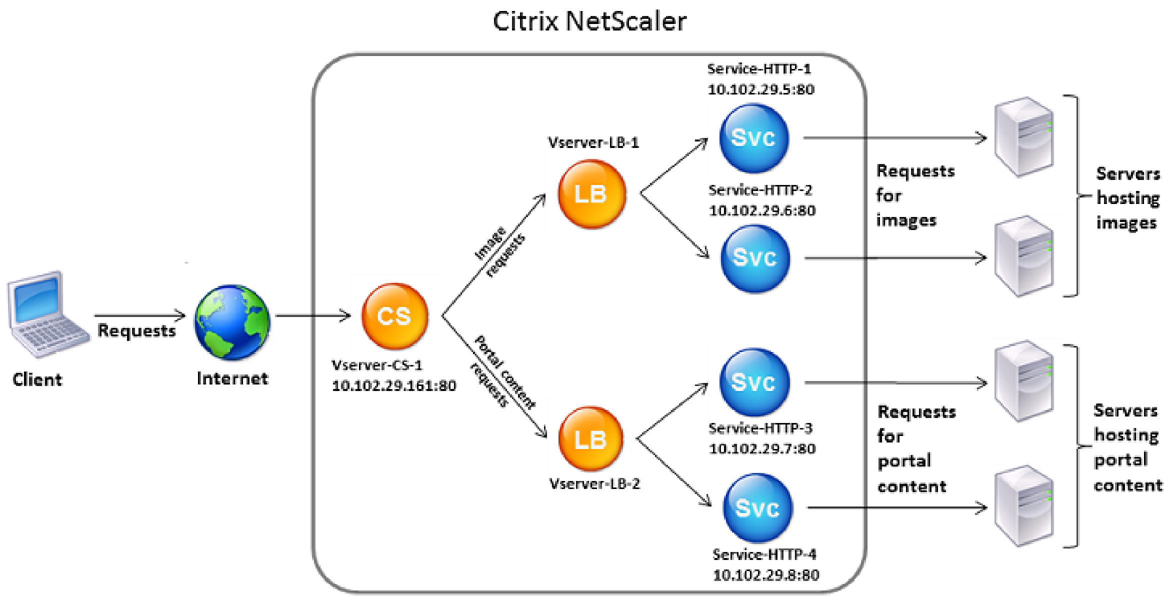
Content Switching enables the NetScaler appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests for static content to another server. You can configure the appliance to perform content switching based on TCP/IP headers and payload.

You can also use content switching to configure the appliance to redirect requests to different servers with different content on the basis of various client attributes. Some of those client attributes are:

- **Device Type.** The appliance examines the user agent or custom HTTP header in the client request for the type of device from which the request originated. Based on the device type, it directs the request to a specific Web server. For example, if the request came from a cell phone, the request is directed to a server that is capable of serving content that the user can view on his or her cell phone. A request from a computer is directed to a different server that is capable of serving content designed for a computer screen.
- **Language.** The appliance examines the Accept-Language HTTP header in the client request and determines the language used by the client's browser. The appliance then sends the request to a server that serves content in that language. For example, using content switching based on language, the appliance can send someone whose browser is configured to request content in French to a server with the French version of a newspaper. It can send someone else whose browser is configured to request content in English to a server with the English version.
- **Cookie.** The appliance examines the HTTP request headers for a cookie that the server set previously. If it finds the cookie, it directs requests to the appropriate server, which hosts custom content. For example, if a cookie is found that indicates that the client is a member of a customer loyalty program, the request is directed to a faster server or one with special content. If it does not find a cookie, or if the cookie indicates that the user is not a member, the request is directed to a server for the general public.
- **HTTP Method.** The appliance examines the HTTP header for the method used, and sends the client request to the right server. For example, GET requests for images can be directed to an image server, while POST requests can be directed to a faster server that handles dynamic content.
- **Layer 3/4 Data.** The appliance examines requests for the source or destination IP, source or destination port, or any other information present in the TCP or UDP headers, and directs the client request to the right server. For example, requests from source IPs that belong to customers can be directed to a custom web portal on a faster server, or one with special content.

A typical content switching deployment consists of the entities described in the following diagram.

Figure 1. Content Switching Architecture


Citrix NetScaler
Client
Requests
Internet
CS
Vserver-CS-1
10.102.29.161:80
Image requests
Portal content requests
LB
Vserver-LB-1
LB
Vserver-LB-2
Svc
Service-HTTP-1
10.102.29.5:80
Svc
Service-HTTP-2
10.102.29.6:80
Svc
Service-HTTP-3
10.102.29.7:80
Svc
Service-HTTP-4
10.102.29.8:80
Requests for images
Requests for portal content
Servers hosting images
Servers hosting portal content

A content switching configuration consists of a content switching virtual server, a load balancing setup consisting of load balancing virtual servers and services, and content switching policies. To configure content switching, you must configure a content switching virtual server and associate it with policies and load balancing virtual servers. This process creates a *content group*—a group of all virtual servers and policies involved in a particular content switching configuration.

Content switching can be used with HTTP, HTTPS, TCP, and UDP connections. For HTTPS, you must enable SSL Offload.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. The priority of the policy defines the order in which the policies bound to the content switching virtual server are evaluated. If you are using default syntax policies, when you bind a policy to the content switching virtual server, you must assign a priority to that policy. If you are using NetScaler classic policies, you can assign a priority to your policies, but are not required to do so. If you assign priorities, the policies are evaluated in the order that you set. If you do not, the NetScaler appliance evaluates your policies in the order in which they were created.

In addition to configuring policy priorities, you can manipulate the order of policy evaluation by using Goto expressions and policy bank invocations. For more details about default syntax policy configuration, see "Configuring Default Syntax Policies."

After it evaluates the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

Content switching virtual servers can only send requests to other virtual servers. If you are using an external load balancer, you must create a load balancing virtual server for it and bind its virtual server as a service to the content switching virtual server.

## Configuring basic content switching

September 21, 2020

Before you configure content switching, you must understand how content switching is set up and how the services and virtual servers are connected.

To configure a basic, functional content switching setup, first enable the content switching feature. Then, create at least one content group. For each content group, create a content switching virtual server to accept requests to a group of web sites that use content switching. Also create a load balancing setup, which includes a group of load balancing virtual servers to which the content switching virtual server directs requests. To specify which requests to direct to which load balancing virtual server, create at least two content switching policies, one for each type of request that is to be redirected. When you have created the virtual servers and policies, bind the policies to the content switching virtual server. You can also bind a policy to multiple content switching virtual servers. When you bind a policy, you specify the load balancing virtual server to which requests that match the policy are to be directed.

In addition to binding individual policies to a content switching virtual server, you can bind policy labels. If you create additional content groups, you can bind a policy or policy label to more than one of the content switching virtual servers.

> **Note**
>
> After creating a content group, you can modify its content switching virtual server to customize the configuration. For information on modifying the configuration of an existing content switching virtual server, see “
> Customizing the Basic Content Switching Configuration.” For information on disabling and reenabling entities, unbinding policies, and removing entities, see “
> Managing a Content Switching Setup.”

This section includes the following details:

- Enabling Content Switching
- Creating Content Switching Virtual Servers
- Configuring a Load Balancing Setup for Content Switching
- Configuring a Content Switching Action
- Configuring Content Switching Policies

- Configuring Content Switching Policy Labels
- Binding Policies to a Content Switching Virtual Server
- Configuring Policy Based Logging for Content Switching
- Verifying the Configuration

## Enabling content switching

September 21, 2020

To use the content switching feature, you must enable content switching. You can configure content switching entities even though the content switching feature is disabled. However, the entities will not work.

### To enable content switching by using the command line interface

At the command prompt, type the following commands to enable content switching and verify the configuration:

- enable ns feature CS
- show ns feature

Example

```
> enable  feature ContentSwitch
 Done
> show feature

        Feature                 Acronym              Status
        -------                 -------              ------
 1)     Web Logging             WL                   OFF
 2)     Surge Protection        SP                   ON
 3)     Load Balancing          LB                   ON
 4)     Content Switching       CS                   ON
 .
 .
 .
 22)    Responder               RESPONDER            ON
 23)    HTML Injection          HTMLInjection        ON
 24)    NetScaler Push          push                 OFF
 Done
<!--NeedCopy-->
```

To enable content switching by using the configuration utility

Navigate to **System > Settings** and, in the Modes and Features group, select **Configure Basic Features**, and select **Content Switching**.

Creating content switching virtual servers

September 21, 2020

You can add, modify, and remove content switching virtual servers. The state of a virtual server is DOWN when you create it, because the load balancing virtual server is not yet bound to it.

To create a virtual server by using the command line interface

At the command prompt, type:

`add cs vserver \<name\> \<protocol\> \<IPAddress\> \<port\>`

Example

```
add cs vserver Vserver-CS-1 HTTP 10.102.29.161 80
<!--NeedCopy-->
```

To add a content switching virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, and add a virtual server.

Configuring a load balancing setup for content switching

September 21, 2020

The content switching virtual server redirects all requests to a load balancing virtual server. You must create one load balancing virtual server for each version of the content that is being switched. This is true even when your setup has only one server for each version of the content, and you are therefore not doing any load balancing with those servers. You can also configure actual load balancing with multiple load-balanced servers that mirror each version of the content. In either scenario, the content switching virtual server needs to have a specific load balancing virtual server assigned to each version of the content that is being switched.

The load balancing virtual server then forwards the request to a service. If it has only one service bound to it, it selects that service. If it has multiple services bound to it, it uses its configured load balancing method to select a service for the request, and forwards that request to the service that it selected.

To configure a basic load balancing setup, you need to perform the following tasks:

- Create load balancing virtual servers
- Create services
- Bind services to the load balancing virtual server

For more information on load balancing, see Load Balancing. For detailed instructions on setting up a basic load balancing configuration, see Set up basic load balancing.

Configuring a content switching action

September 21, 2020

You specify the target load balancing virtual server for a content switching policy when binding the policy to the content switching virtual server. Consequently, you have to configure one policy for each load balancing virtual server to which to direct traffic.

However, if your content switching policy uses a default syntax rule, you can configure an action for the policy. In the action, you can specify the name of the target load balancing virtual server, or you can configure a request-based expression that, at run time, computes the name of the load balancing virtual server to which to send the request. The action expression must be specified in the default syntax.

The expression option can drastically reduce the size of your content switching configuration, because you need only one policy per content switching virtual server. Content switching policies that use an action can also be bound to multiple content switching virtual servers, because the target load balancing virtual server is no longer specified in the content switching policy. The ability to bind a single policy to multiple content switching virtual servers helps to further reduce the size of your content switching configuration.

After you create an action, you create a content switching policy and specify the action in the policy, so that the action is performed when that policy matches a request.

> **Note**
>
> You can also, for a content switching policy that uses a default syntax rule, specify the target load balancing virtual server when binding the policy to a content switching virtual server, instead of using a separate action. For domain-based policies, URL-based policies, and rule based policies that use classic expressions, an action is not available. So, for these types of policies, you specify the name of the target load balancing virtual server when binding the policy to a content switch-

> ing virtual server. For more information, see “ Binding Policies to a Content Switching Virtual Server.”

Configuring an Action that Specifies the Name of the Target Load Balancing Virtual Server

If you choose to specify the name of the target load balancing virtual server in a content switching action, you need as many content switching policies as you have target load balancing virtual servers. Content switching decisions, in this case, are based on the rule in the content switching policy, and the action merely specifies the target load balancing virtual server. When a request matches the policy, the request is forwarded to the specified load balancing virtual server.

To create and verify a content switching action that specifies the name of the target load balancing virtual server, by using the command line interface

At the command prompt, type:

- add cs action <name> -targetLBVserver <string> [-comment <string>]
- show cs action <name>

Example

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "Forwards requests to mylbvserver."
 Done
> show cs action mycsaction
    Name: mycsaction
    Target LB Vserver: mylbvserver
    Hits: 0
    Undef Hits: 0
    Action Reference Count: 0
    Comment: "Forwards requests to mylbvserver."

 Done
>
<!--NeedCopy-->
```

To configure a content switching action that specifies the name of the target load balancing virtual server, by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Actions**.

2. Configure a content switching action, and specify the name of the target load balancing virtual server.

Configuring an Action that Specifies an Expression for Selecting the Target at Run Time

If you choose to configure a request-based expression that can dynamically compute the name of the target load balancing virtual server, you need to configure only one content switching policy to select the appropriate virtual server. The rule for the policy can be a simple TRUE (the policy matches all requests) because, in this case, content switching decisions are based on the expression in the action. By configuring an expression in an action, you can drastically reduce the size of your content switching configuration.

If you choose to configure a request-based expression for computing the name of the target load balancing virtual server at run time, you must carefully consider how to name the load balancing virtual servers in the configuration. You must be able to derive their names by using the request-based policy expression in the action.

For example, if you are switching requests on the basis of the URL suffix (file extension of the requested resource), when naming the load balancing virtual servers, you can follow the convention of appending the URL suffix to a predetermined string, such as mylb_. For example, load balancing virtual servers for HTML pages and PDF files could be named mylb_html and mylb_pdf, respectively. In that case, the rule that you can use in the content switching action, to select the appropriate load balancing virtual server, is "mylb_"+HTTP.REQ.URL.SUFFIX. If the content switching virtual server receives a request for an HTML page, the expression returns mylb_html, and the request is switched to virtual server mylb_html.

To create a content switching action that specifies an expression, by using the command line interface

At the command line, type the following commands to create a content switching action that specifies an expression and verify the configuration:

```
-  add cs action \<name\> -targetVserverExpr \<expression\>) \[-comment \<string\>\]
-  show cs action \<name\>
<!--NeedCopy-->
```

Example

```
> add cs action mycsaction1 -targetvserverExpr '"mylb_" + HTTP.REQ.URL.SUFFIX'
 Done
```

```
> show cs action mycsaction1
    Name: mycsaction1
    Target Vserver Expression: "mylb_" + HTTP.REQ.URL.SUFFIX
    Target LB Vserver: No_Target
               …
 Done
>
<!--NeedCopy-->
```

To configure a content switching action that specifies an expression by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Actions**.
2. Configure a content switching action, and specify an expression that will dynamically compute the name of the target load balancing virtual server.

Configuring content switching policies

September 21, 2020

A content switching policy defines a type of request that is to be directed to a load balancing virtual server. These policies are applied in the order of the priorities assigned to them or (if you are using NetScaler appliance classic policies and do not assign priorities when binding them) in the order in which the policies were created.

The policies can be:

- **Domain-based policies.** The NetScaler appliance compares the domain of an incoming URL with the domains specified in the policies. The appliance then returns the most appropriate content. Domain-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.
- **URL-based policies.** The appliance compares an incoming URL with the URLs specified in the policies. The appliance then returns the most appropriate URL-based content, which is usually the longest matching configured URL. URL-based policies must be classic policies; default syntax policies are not supported for this type of content switching policy.
- **Rule-based policies.** The appliance compares incoming data to expressions specified in the policies. You create rule-based policies by using either a classic expression or a default syntax expression. Both classic and default syntax policies are supported for rule-based content switching policies.

> **Note**
>
> A rule based policy can be configured with an optional action. A policy with an action can be bound to multiple virtual servers or policy labels.

If you set a priority when binding your policies to the content switching virtual server, the policies are evaluated in order of priority. If you do not set specific priorities when binding your policies, the policies are evaluated in the order in which they were created.

For information about NetScaler appliance classic policies and expressions, see "Configuring Classic Policies and Expressions." For information about Default Syntax policies, see "Configuring Default Syntax Expressions."

The following steps help you to bind an authentication profile to a content switching virtual server:

- Create an authentication profile. For more information, see Creating an authentication profile.
- Bind the authentication profile to the content switching by using the following command.
  - `set cs vserver <csname> -authnProfile <authnProfileName>`

To create a content switching policy by using the command line interface

At the command prompt, type one of the following commands:

- `add cs policy <policyName> -domain <domain>`
- `add cs policy <policyName> -url <URLValue>`
- `add cs policy <policyName> -rule <RULEValue>`
- `add cs policy <policyName> -rule <RULEValue> -action <actionName>`

Example:

```
add cs policy Policy-CS-1 -url "http://example.com"

add cs policy Policy-CS-1 -domain "example.com"

add cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(24).EQ
    (10.217.84.0)"

add cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2009 Nov,GMT 2009
     Dec)"

add cs policy Policy-CS-3 -rule "http.req.method.eq(GET)" -action act1
<!--NeedCopy-->
```

To rename a content switching policy by using the command line interface

At the command prompt, type:

`rename cs policy <policyName> <newName>`

Example:

```
rename cs policy myCSPolicy myCSPolicy1
<!--NeedCopy-->
```

To rename a content switching policy by using the configuration utility

Navigate to **Traffic Management > Content Switching > Policies**, select a policy and, in the Action list, select Rename.

To create a content switching policy by using the configuration utility

Navigate to **Traffic Management > Content Switching > Policies**, and configure a content switching policy.

Configuring content switching policy labels

September 21, 2020

A policy label is a user-defined bind point to which policies are bound. When a policy label is invoked, all the policies bound to it are evaluated in the order of the priority that you assigned to them. A policy label can include one or more policies, each of which can be assigned its own result. A match on one policy in the policy label can result in proceeding to the next policy, invoking a different policy label or appropriate resource, or an immediate end to policy evaluation and return of control to the policy that invoked the policy label. You can create policy labels for default syntax policies only.

For information about policy labels, see the "Creating Policy Labels."

A content switching policy label consists of a name, a label type, and a list of policies bound to the policy label. The policy label type specifies the protocol that was assigned to the policies bound to the label. It must match the service type of the content switching virtual server to which the policy that invokes the policy label is bound. For example, you can bind TCP Payload policies to a policy label of type TCP only. Binding TCP Payload policies to a policy label of type HTTP is not supported.

Each policy in a content switching policy label is associated with either a target (which is equivalent to the action that is associated with other types of policies, such as rewrite and responder policies) or a gotoPriorityExpression option and/or an invoke option. That is, for a given policy in a content switching policy label, you can specify a target, or you can set the gotoPriorityExpression option and/or the invoke option. Additionally, if multiple policies evaluate to true, only the target of the last policy that evaluates to true is considered.

You can use either the NetScaler appliance CLI or the configuration utility to configure content switching policy labels. In the NetScaler appliance CLI, you first create a policy label by using the add cs policylabel command. Then, you bind policies to the policy label, one policy at a time, by using the bind cs policylabel command. In the NetScaler appliance GUI, you perform both tasks in a single dialog box.

To create a content switching policy label by using the command line interface

At the command prompt, type:

add cs policylabel <labelName> <cspolicylabelTypetype>

Example

```
add cs policylabel testpollab http
<!--NeedCopy-->
```

To rename a content switching policy label by using the command line interface

At the command prompt, type:

rename cs policylabel <labelName> <newName>

Example

```
rename cs policylabel oldPolicyLabelName newPolicyLabelName
<!--NeedCopy-->
```

To rename a content switching policy label by using the configuration utility

Navigate to Traffic Management > Content Switching > Policy Labels , select a policy label and, in the Action list, select Rename.

To bind a policy to a content switching policy label by using the command line interface

At the command prompt, type the following commands to bind a policy to a policy label and verify the configuration:

bind cs policylabel <labelName> <policyName> <priority>[[-targetVserver <string>]	[-gotoPriorityExpression <expression>]	[-invoke <labeltype> <labelName>]]

-
- show cs policylabel <labelName>

Example

```
bind cs policylabel testpollab test_Pol 100 -targetVserver LBVIP
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 1
        Number of times invoked: 0
        Policy Name: test_Pol
        Priority: 100
        Target Virtual Server: LBVIP
<!--NeedCopy-->
```

> **Note:** If a policy is configure with an action, the target virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke (invoke) parameters are not required. If a policy is not configure with an action, you need to configure at least one of the following parameters: targetVserver, gotoPriorityExpression, and invoke.

To unbind a policy from a policy label by using the command line interface

At the command prompt, type the following commands to unbind a policy from a policy label and verify the configuration:

- unbind cs policylabel <labelName> <policyName>
- show cs policylabel <labelName>

Example

```
unbind cs policylabel testpollab test_Pol
show cs policylabel testpollab
        Label Name: testpollab
        Label Type: HTTP
        Number of bound policies: 0
        Number of times invoked: 0
<!--NeedCopy-->
```

To remove a policy label by using the command line interface

At the command prompt, type:

rm cs policylabel <labelName>

To manage a content switching policy label by using the configuration utility

Navigate to **Traffic Management > Content Switching > Policy Labels**, configure a policy label, bind policies to the label, and optionally specify a priority, gotoPriority expression, and an invoke option.

Binding policies to a content switching virtual server

September 21, 2020

After you create your content switching virtual server and policies, you bind each policy to the content switching virtual server. When binding the policy to the content switching virtual server, you specify the target load balancing virtual server.

> **Note**
>
> If your content switching policy uses a default syntax rule, you can configure a content switching action for the policy. If you configure an action, you must specify the target load balancing virtual server when you are configuring the action, not when you are binding the policy to the content switching virtual server. For more information about configuring a content switching action, see Configuring a Content Switching Action.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the command line interface

At the command prompt, type:

```
bind cs vserver \<name\>\[-lbvserver\<string\> -targetLBVServer\<string
    \> -policyname \<string\> -priority \<positive\_integer\>\] \[-
    gotoPriorityExpression \<expression\>\] \[-type ( REQUEST | RESPONSE
     )\] \[-invoke (\<labelType\> \<labelName\>) \]
<!--NeedCopy-->
```

Example:

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -
    gotoPriorityExpression NEXT

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -
    gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1
<!--NeedCopy-->
```

```
bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -
    priority 20
<!--NeedCopy-->
```

> **Note**
>
> The parameters, target load balancing virtual server (targetVserver), go to priority expression (gotoPriorityExpression), and invoke method (invoke) cannot be used if a policy has an action.

To bind a policy to a content switching virtual server and select a target load balancing virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, open a virtual server and, in the
Content Switching Policy Binding section, bind a policy to the virtual server, and specify a target load balancing virtual server.

Configuring policy based logging for content switching

September 21, 2020

You can configure policy based logging for a content switching policy. Policy based logging enables you to specify a format for log messages. The contents of the log message are defined by using a default syntax expression in the content switching policy. When the content switching action specified in the policy is performed, the NetScaler appliance constructs the log message from the expression and writes the message to the log file. Policy based logging is particularly useful if you want to test and troubleshoot a configuration in which content switching actions identify the target load balancing virtual server at run time.

> **Note**
>
> If multiple policies bound to a given virtual server evaluate to TRUE and are configured with an audit message action, the NetScaler appliance does not perform all the audit message actions. It performs only the audit message action that is configured for the policy whose content switching action is performed.

To configure policy based logging for a content switching policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configuring the NetScaler appliance for Audit Logging. After you configure the audit message action, you specify the action in a content switching policy.

To configure policy based logging for a content switching policy by using the command line interface

At the command line, type the following commands to configure policy based logging for a content switching policy and verify the configuration:

- set cs policy <policyName> -logAction <string>
- show cs policy <policyName>

Example

```
> set cs policy cspol1 -logAction csLogAction
 Done
> show cs policy cspol1

    Policy: cspol1  Rule: TRUE  Action: csact1
    LogAction: csLogAction
    Hits: 0

1)  CS Vserver: csvs1
    Priority: 10
 Done
>
<!--NeedCopy-->
```

To configure policy based logging for a content switching policy by using the configuration utility

Navigate to **Traffic Management > Content Switching > Policies**, open a policy and, in the Log Action list, select a log action for the policy.

Verifying the configuration

September 21, 2020

To verify that your content switching configuration is correct, you need to view the content switching entities. To verify proper operation after your content switching configuration has been deployed, you can view the statistics that are generated as the servers are accessed.

Viewing the Properties of Content Switching Virtual Servers

You can view the properties of content switching virtual servers that you have configured on the NetScaler appliance. You can use the information to verify whether the virtual server is correctly configured and, if necessary, to troubleshoot. In addition to details such as name, IP address, and port, you can view the various policies bound to a virtual server, and its traffic-management settings.

The content switching policies are displayed in the order of their priority. If more than one policy has the same priority, they are shown in the order in which they are bound to the virtual server.

> **Note**
>
> If you have configured the content switching virtual server to forward traffic to a load balancing virtual server, you can also view the content switching policies by viewing the properties of the load balancing virtual server.

To view the properties of content switching virtual servers by using the command line interface

To list basic properties of all content switching virtual servers in your configuration, or detailed properties of a specific content switching virtual server, at the command prompt, type one of the following commands:

- `show cs vserver`
- `show cs vserver \<name\>`

Example

```
1.
show cs vserver Vserver-CS-1
Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
Last state change was at Thu Jun 30 10:48:59 2011
Time since last state change: 6 days, 20:03:00.760
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
Default: Content Precedence: RULE
Vserver IP and Port insertion: OFF
Case Sensitivity: ON
Push: DISABLED Push VServer:
Push Label Rule: none

...
1) Policy : __ESNS_PREBODY_POLICY Priority:0
2) Policy : __ESNS_POSTBODY_POLICY Priority:0

1) Compression Policy Name: __ESNS_CMP_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

2) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

3) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

4) Responder Policy Name: __ESNS_RESPONDER_POLICY Priority: 2147483647
GotoPriority Expression: END

1) Policy: wiki Target: LBVIP2 Priority: 25 Hits: 0
2) Policy: plain Target: LBVIP1 Priority: 90 Hits: 0
3) Policy: DispOrderTest2 Target: KerbAuthLBVS Priority: 91 Hits: 0
4) Policy: test_Pol Target: LBVIP1 Priority: 92 Hits: 0
5) Policy: PolicyNameTesting Target: LBVIP1 Priority: 100 Hits: 0
 Done
>
```

```

show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
…
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
…
Client Idle Timeout: 180 sec
Down state flush: DISABLED
…

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
…
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
…
<!--NeedCopy-->
```

Viewing Content Switching Policies

You can view the properties of the content switching policies that you defined, such as the name, domain, and URL or expression, and use the information to find any mistakes in the configuration, or to troubleshoot if something is not working as it should.

To view the properties of content switching policies by using the command line interface

To list either basic properties of all content switching policies in your configuration or detailed properties of a specific content switching policy, at the command prompt, type one of the following commands:

- `show cs policy`
- `show cs policy \<PolicyName\>`

Example

```
show cs policy

show cs policy Policy-CS-1
<!--NeedCopy-->
```

To view the properties of content switching policies by using the configuration utility

Navigate to **Traffic Management > Content Switching > Policies**, select a policy and, in the Action list, select Show Bindings.

Viewing a Content Switching Virtual Server Configuration by Using the Visualizer

The Content Switching Visualizer is a tool that you can use to view a content switching configuration in graphical format. You can use the visualizer to view the following configuration items:

- A summary of the load balancing virtual servers to which the content switching virtual server is bound.
- All services and service groups that are bound to the load balancing virtual server and all monitors that are bound to the services.
- The configuration details of any displayed element.
- Any policies bound to the content switching virtual server. These policies need not be content switching policies. Many types of policies, such as Rewrite policies, can be bound to a content switching virtual server.

After you configure the various elements in a content switching and load balancing setup, you can export the entire configuration to an application template file.

> **Note**
>
> The Visualizer requires a graphical interface, so it is available only through the configuration utility.

To view a content switching configuration by using the Visualizer in the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Content Switching Visualizer window, you can adjust the viewable area as follows:
    - Click the Zoom In and Zoom Out icons to increase or decrease the viewable area.
    - Click the Save Image icon to save the graph as an image file.
    - In the Search in text field, begin typing the name of the item you are looking for. When you have typed enough characters to identify the item, its location is highlighted. To restrict

the search, click the drop-down menu and select the type of element that you want to search for.

4. To view configuration details for entities that are bound to this virtual server, you can do the following:
    - To view policies that are bound to the virtual server, in the tool bar at the top of the dialog box select one or more feature-specific policy icons. If policy labels are configured, they appear in the main view area.
    - To view the configuration details for a bound service or service group, click the icon for the service, click the Related Tasks tab, and then click Show Member Services.
    - To view the configuration details for a monitor, click the icon for the monitor, click the Related Tasks tab, and then click View Monitor.
5. To view detailed statistics for any virtual server in the content switching configuration, click the virtual server for which you want to view statistics, then click the Related Tasks tab, and then click Statistics.
6. To view a comparative list of the parameters whose values either differ or are not defined across service containers for a load balancing virtual server, click the icon for a container, click the Related Tasks tab, and then click Service Attributes Diff.
7. To view monitor binding details for the services in a container, in the Service Attributes Diff dialog box, in the Group column for the container, click Details. This comparative list helps you determine which service container has the configuration you want to apply to all the service containers.
8. To view the number of requests received per second at a given point in time by the virtual servers in the configuration, and the number of hits per second at a given point in time for rewrite, responder, and cache policies, click Show Stats. The statistical information is displayed on the respective nodes in the Visualizer. This information is not updated in real time. It has to be refreshed manually. To refresh the information, click Refresh Stats.
    Note: This option is available only on NetScaler appliance nCore builds.
9. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click Related Tasks, click Copy Properties, and then paste the information into a document.
10. To export the entire configuration that is displayed in the Visualizer to an application template file, click the icon for the content switching virtual server, click Related Tasks, and then click Create Template. When creating the application template, you can configure variables in some policy expressions and actions. For more information about creating the application template file and configuring variables for a template, see AppExpert.

Customizing the basic content switching configuration

September 21, 2020

After you configure a basic content switching setup, you might need to customize it to meet your requirements. If your web servers are UNIX-based and rely on case sensitive pathnames, you can configure case sensitivity for policy evaluation. You can also set precedence for evaluation of the content switching policies that you configured. You can configure HTTP and SSL content switching virtual servers to listen on multiple ports instead of creating separate virtual servers. If you want to configure content switching for a specific a virtual LAN, you can configure a content switching virtual server with a listen policy.

Configuring case sensitivity for policy evaluation

You can configure the content switching virtual server to treat URLs as case sensitive in URL-based policies. When case sensitivity is configured, the NetScaler appliance considers case when evaluating policies. For example, if case sensitivity is off, the URLs /a/1.htm and /A/1.HTM are treated as identical. If case sensitivity is on, those URLs are treated as separate and can be switched to different targets.

To configure case sensitivity by using the command line interface

At the command prompt, type:

`set cs vserver \<name\> -caseSensitive (ON|OFF)`

Example:

```
set cs vserver Vserver-CS-1 -caseSensitive ON
<!--NeedCopy-->
```

To configure case sensitivity by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and then select Case Sensitive.

Setting the precedence for policy evaluation

Precedence refers to the order in which policies that are bound to a virtual server are evaluated. You do not normally have to configure precedence: the default precedence works correctly in many cases. If you want to make sure that one policy or set of policies is applied first, however, and another policy or set of policies is applied only if the first set does not match a request, you can configure either URL-based precedence or rule-based precedence.

Precedence with URL-Based policies

If there are multiple matching URLs for the incoming request, the precedence (priority) for URL-based policies is:

1. Domain and exact URL
2. Domain, prefix, and suffix
3. Domain and suffix
4. Domain and prefix
5. Domain only
6. Exact URL
7. Prefix and suffix
8. Suffix only
9. Prefix only
10. Default

If you configure precedence based on URL, the request URL is compared to the configured URLs. If none of the configured URLs match the request URL, then rule-based policies are checked. If the request URL does not match any rule-based policies, or if the content group selected for the request is down, then the request is processed as follows:

- If you configure a default group for the content switching virtual server, then the request is forwarded to the default group.
- If the configured default group is down or if no default group is configured, then an "HTTP 404 Not Found" error message is sent to the client.

> **Note**
>
> You should configure URL-based precedence if the content type (for example, images) is the same for all clients. However, if different types of content must be served based on client attributes (such as Accept-Language), you must use rule-based precedence.

Precedence with Rule-Based policies

If you configure precedence based on rules, which is the default setting, the request is tested on the basis of the rule-based policies you have configured. If the request does not match any rule-based policies, or if the content group selected for the incoming request is down, the request is processed in the following manner:

- If a default group is configured for the content switching virtual server, the request is forwarded to the default group.

- If the configured default group is down or if no default group is configured, an "HTTP 404 Not Found" error message is sent to the client.

To configure precedence by using the command line interface

At the command prompt, type:

`set cs vserver \<name\> -precedence ( RULE | URL )`

Example:

```
set cs vserver Vserver-CS-1 -precedence RULE
<!--NeedCopy-->
```

To configure precedence by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Traffic Settings**, and then specify Precedence.

Support for multiple ports for HTTP and SSL type content switching virtual servers

You can configure the NetScaler appliance so that HTTP and SSL content switching virtual servers listen on multiple ports, without having to configure separate virtual servers. This feature is especially useful if you want to base a content switching decision on a part of the URL and other L7 parameters. Instead of configuring multiple virtual servers with the same IP address and different ports, you can configure one IP address and specify the port as *. As a result, the configuration size is also reduced.

To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the command line

At the command prompt, type:

add cs vserver <name> <serviceType> <IPAddress> Port *

Example

```
> add cs vserver cs1 HTTP 10.102.92.215 *
 Done
> sh cs vserver cs1
        cs1 (10.102.92.215:*) - HTTP    Type: CONTENT
        State: UP
        Last state change was at Tue May 20 01:15:49 2014
        Time since last state change: 0 days, 00:00:03.270
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        Port Rewrite : DISABLED
        State Update: DISABLED
        Default:        Content Precedence: RULE
        Vserver IP and Port insertion: OFF
        L2Conn: OFF     Case Sensitivity: ON
        Authentication: OFF
        401 Based Authentication: OFF
        Push: DISABLED  Push VServer:
        Push Label Rule: none
        IcmpResponse: PASSIVE
        RHIstate:  PASSIVE
        TD: 0
 Done
<!--NeedCopy-->
```

To configure an HTTP or SSL content switching virtual server to listen on multiple ports by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and create a virtual server of type HTTP or SSL.
2. Use an asterisk (*) to specify the port.

Configuring per-VLAN Wildcarded virtual servers

If you want to configure content switching for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the command line interface

At the command prompt, type:

- add cs vserver <name> <serviceType> IPAddress `*` `Port` `*` -listenpolicy <expression> [-listenpriority <positive_integer>]

Example:

```
add cs vserver Vserver-CS-vlan1 ANY * *
     -listenpolicy "CLIENT.VLAN.ID.EQ(2)"  -listenpriority 10
<!--NeedCopy-->
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, and configure a virtual server. Specify a listen policy that restricts it to processing traffic only on the specified VLAN.

After you have created this virtual server, you bind it to one or more services as described in Binding Services to the Virtual Server.

Configuring the Microsoft SQL server version setting

You can specify the version of Microsoft® SQL Server® for a content switching virtual server that is of type MSSQL. The version setting is recommended if you expect some clients to not be running the same version as your Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL Server version parameter by using the command line interface

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a content switching virtual server and verify the configuration:

- `set cs vserver \<name\> -mssqlServerVersion \<mssqlServerVersion\>`
- `show cs vserver \<name\>`

Example

```
> set cs vserver myMSSQLcsvip -mssqlServerVersion 2008R2 Done > show cs vserver myMSSQLcsvip myMSSQLcsvip (192.0.2.13:1433) - MSSQL Type: CONTENT State: UP . . . . . . MSsql Server Version: 2008R2 . . . . . . Done >
<!--NeedCopy-->
```

To set the Microsoft SQL server version parameter by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, configure a virtual server and specify the protocol as MYSQL.

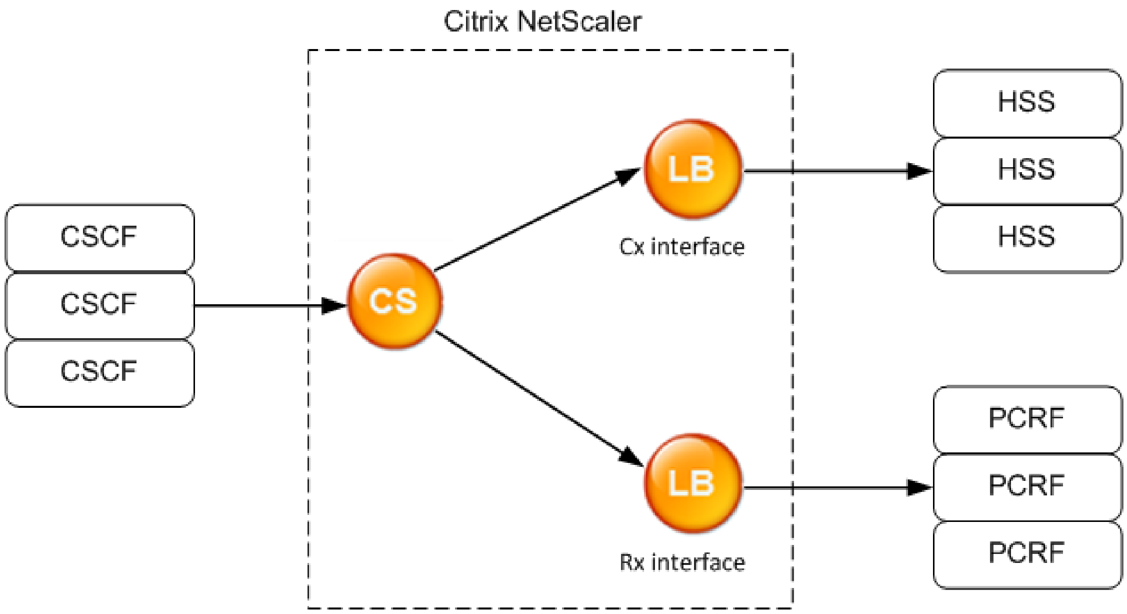
2. In **Advanced Settings**, select MySQL, and specify the Server Version.

Content switching for diameter protocol

September 21, 2020

For Diameter-protocol traffic, you can configure the NetScaler appliance (or virtual appliance) to act as a relay agent that load balances and forwards a packet to the appropriate destination on the basis of the message content (AVP value in the message). Since the appliance does not perform any application-level processing, it provides relaying services for all diameter applications as specified by the configured content switching policies. Therefore, the appliance advertises the Relay Application ID in the capability exchange answer (CEA) message when the client establishes a diameter connection. You must configure a content switching virtual server, load balancing virtual servers, and services to represent the diameter nodes. When a request reaches the content switching virtual server, the virtual server applies the content switching policies associated with that type of request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

A diameter interface provides a connection between the different diameter nodes. The following sample deployment uses Cx and Rx interfaces. A Cx interface provides a connection between a CSCF and an HSS. An Rx interface provides a connection between a CSCF and a PCRF. All the messages reach the NetScaler appliance. Depending on whether the message is for a Cx or an Rx interface, and on the content switching policies defined, the NetScaler appliance selects an appropriate load balancing server pool.



CSCF=Call Session Control Function
HSS=Home Subscriber Server
PCRF=Policy and Charging Rules Function

Sample Configuration

1. For each entity, create a service, a load balancing server, and bind the service to the virtual server.

```
add service svc_pcrf[1-3] 1.1.1.1[1-3] DIAMETER 3868
add service svc_hss[1-3] 1.1.1.2[1-3] DIAMETER 3868
add lb vserver vs_rx DIAMETER -persistenceType DIAMETER -persistavpno 263
add lb vserver vs_cx DIAMETER -persistenceType DIAMETER -persistavpno 263
bind lb vserver vs_rx svc_pcrf[1-3]
bind lb vserver vs_cx svc_hss[1-3]
<!--NeedCopy-->
```

2. Create a content switching virtual server and two actions (one for each load balancing virtual server). Create two content switching policies and bind these policies to the content switching virtual server, specifying a priority for each policy.

```
add cs vserver cs_diameter DIAMETER 10.1.1.10 3868
add cs action cx_action -targetLBVserver vs_cx
add cs action rx_action  - targetLBvserver vs_rx
add cs policy cx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777216)" -action cx_action
add cs policy rx_policy -rule "DIAMETER.REQ.AUTH_APPLICATION_ID.EQ
    (16777236)" -action rx_action
bind cs vserver cs_diameter -policyName rx_policy -priority 100
bind cs vserver cs_diameter -policyName cx_policy -priority 110
<!--NeedCopy-->
```

Protecting the content switching setup against failure

September 21, 2020

Content switching may fail when the content switching virtual server goes DOWN or fails to handle excessive traffic, or for other reasons. To reduce the chances of failure, you can take the following measures to protect the content switching setup against failure:

Configuring a Backup Virtual Server

If the primary content switching virtual server is marked DOWN or DISABLED, the NetScaler appliance can direct requests to a backup content switching virtual server. It can also send a notification message to the client regarding the site outage or maintenance. The backup content switching virtual server is a proxy and is transparent to the client.

When configuring the backup virtual server, you can specify the configuration parameter Disable Primary When Down to ensure that, when the primary virtual server comes back up, it remains the secondary until you manually force it to take over as the primary. This is useful if you want to ensure that any updates to the database on the server for the backup are preserved, enabling you to synchronize the databases before restoring the primary virtual server.

You can configure a backup content switching virtual server when you create a content switching virtual server or when you change the optional parameters of an existing content switching virtual server. You can also configure a backup content switching virtual server for an existing backup content switching virtual server, thus creating cascaded backup content switching virtual servers. The maximum depth of cascaded backup content switching virtual servers is 10. The appliance searches for a backup content switching virtual server that is up and accesses that content switching virtual server to deliver the content.

> **Note:** If a content switching virtual server is configured with both a backup content switching virtual server and a redirect URL, the backup content switching virtual server takes precedence over the redirect URL. The redirect is used when the primary and backup virtual servers are down.

To set up a backup content switching virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver <name> -backupVserver <string> -disablePrimaryOnDown (ON|OFF)
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -disablePrimaryOnDown ON
<!--NeedCopy-->
```

To set up a backup content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and specify a Backup Virtual Server.

Diverting Excess Traffic to a Backup Virtual Server

The spillover option diverts new connections arriving at a content switching virtual server to a backup content switching virtual server when the number of connections to the content switching virtual server exceeds the configured threshold value. The threshold value is dynamically calculated, or you can set the value. The number of established connections (in case of TCP) at the virtual server is compared with the threshold value. When the number of connections reaches the threshold, new connections are diverted to the backup content switching virtual server.

If the backup content switching virtual servers reach the configured threshold and are unable to take the load, the primary content switching virtual server diverts all requests to the redirect URL. If a redirect URL is not configured on the primary content switching virtual server, subsequent requests are dropped.

To configure a content switching virtual server to divert new connections to a backup virtual server by using the command line interface

At the command prompt, type:

```
set cs vserver \<name\> -soMethod \<methodType\> -soThreshold \<thresholdValue\> -soPersistence \<persistenceValue\> -soPersistenceTimeout \<timeoutValue\>
```

```
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

To set a content switching virtual server to divert new connections to a backup virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and configure spillover.

Configuring a Redirection URL

You can configure a redirect URL to communicate the status of the NetScaler appliance in the event that a content switching virtual server of type HTTP or HTTPS is DOWN or DISABLED. This URL can be local or remote.

Redirect URLs can be absolute URLs or relative URLs. If the configured redirect URL contains an absolute URL, the HTTP redirect is sent to the configured location, regardless of the URL specified in the incoming HTTP request. If the configured redirect URL contains only the domain name (relative URL), the HTTP redirect is sent to a location after appending the incoming URL to the domain configured in the redirect URL.

Citrix recommends using an absolute URL. That is, a URL ending in /, for example www.example.com/ instead of a relative URL. A relative URL redirection might result in the vulnerability scanner reporting a false positive.

Note: If a content switching virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect URL is used when the primary and backup virtual servers are down.

When redirection is configured and the content switching virtual server is unavailable, the appliance issues an HTTP 302 redirect to the user's browser.

To configure a redirect URL for when the content switching virtual server is unavailable by using the command line interface

At the command prompt, type:

```
set cs vserver \<name\> -redirectURL \<URLValue\>
<!--NeedCopy-->
```

Example

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/
    mysite/maintenance
<!--NeedCopy-->
```

To configure a redirect URL for when the content switching virtual server is unavailable by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Protection, and specify a Redirect URL.

Configuring the State Update Option

The content switching feature enables the distribution of client requests across multiple servers on the basis of the specific content presented to the users. For efficient content switching, the content switching virtual server distributes the traffic to the load balancing virtual servers according to the content type, and the load balancing virtual servers distribute the traffic to the physical servers according to the specified load balancing method.

For smooth traffic management, it is important for the content switching virtual server to know the status of the load balancing virtual servers. The state update option helps to mark the content switching virtual server DOWN if the load balancing virtual server bound to it is marked DOWN. A load balancing virtual server is marked DOWN if all the physical servers bound to it are marked DOWN.

When State Update is disabled:

The status of the content switching virtual server is marked as UP. It remains UP even if there is no bound load balancing virtual server that is UP.

When State Update is enabled:

When you add a new content switching virtual server, initially, its status is shown as DOWN. When you bind a load balancing virtual server whose status is UP, the status of the content switching virtual server becomes UP.

If more than one load balancing virtual server is bound and if one of them is specified as the default, the status of the content switching virtual server reflects the status of the default load balancing virtual server.

If more than one load balancing virtual server is bound without any of them being specified as the default, the status of the content switching virtual server is marked UP only if all the bound load balancing virtual servers are UP.

To configure the state update option by using the command line interface

At the command prompt, type:

```
add cs vserver \<name\> \<protocol\> \<ipAddress\> \<port\> -stateUpdate ENABLED
<!--NeedCopy-->
```

Example

```
 add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED -cltTimeout 180
<!--NeedCopy-->
```

To configure the state update option by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, configure a virtual server and specify the protocol as MYSQL.
2. In Advanced Settings, select Traffic Settings, and then select State Update.

Flushing the Surge Queue

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

> Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.
> Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the command line interface

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<serverName> and <port>) without specifying the name of the service group (<name>) and you cannot specify <port> without a <serverName>. Specify the <serverName> and <port> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>].
<!--NeedCopy-->
```

Examples

```
1. flush ns surgeQ -name SVC1ANZGB -serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

To flush a surge queue by using the configuration utility

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

Managing a content switching setup

September 21, 2020

After a content switching setup is configured, it may require periodic changes. When operating systems or software are updated, or hardware wears out and is replaced, you may need to take down your setup. Load on your setup may increase, requiring additional resources. You may also modify the configuration to improve performance.

These tasks may require unbinding policies from the content switching virtual server, or disabling or removing content switching virtual servers. After you have made changes to your setup, you may need to re-enable servers and rebind policies. You might also want to rename your virtual servers.

Unbinding Policies from the Content Switching Virtual Server

When you unbind a content switching policy from its virtual server, the virtual server no longer includes that policy when determining where to direct requests.

To unbind a policy from a content switching virtual server by using the command line interface

At the command prompt, type:

`unbind cs vserver <name> -policyname <string>`

Example:

```
unbind cs vserver Vserver-CS-1 -policyname Policy-CS-1
<!--NeedCopy-->
```

To unbind a policy from a content switching virtual server by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Virtual Servers**, and open the virtual server.
2. Click in the **Policies** section, select the policy, and click **Unbind**.

Removing Content Switching Virtual Servers

You normally remove a content switching virtual server only when you no longer require the virtual server. When you remove a content switching virtual server, the NetScaler appliance first unbinds all policies from the content switching virtual server, and then removes it.

To remove a content switching virtual server by using the command line interface

At the command prompt, type:

`rm cs vserver <name>`

Example:

```
rm cs vserver Vserver-CS-1
<!--NeedCopy-->
```

To remove a content switching virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server, and click **Delete**.

Disabling and Re-Enabling Content Switching Virtual Servers

Content switching virtual servers are enabled by default when you create them. You can disable a content switching virtual server for maintenance. If you disable the content switching virtual server, the state of the content switching virtual server changes to Out of Service. While out of service, the content switching virtual server does not respond to requests.

To disable or re-enable a virtual server by using the command line interface

At the command prompt, type one of the following commands:

- `disable cs vserver <name>`
- `enable cs vserver <name>`

Example:

```
disable cs vserver Vserver-CS-1

enable cs vserver Vserver-CS-1
<!--NeedCopy-->
```

To disable or re-enable a virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server and, in the Action list, select Enable or Disable.

Renaming Content Switching Virtual Servers

You can rename a content switching virtual server without unbinding it. The new name is propagated automatically to all affected parts of the NetScaler appliance configuration.

To rename a virtual server by using the command line interface

At the command prompt, type:

`rename cs vserver <name> <newName>`

Example:

```
rename cs vserver Vserver-CS-1 Vserver-CS-2
<!--NeedCopy-->
```

To rename a virtual server by using the configuration utility

Navigate to **Traffic Management > Content Switching > Virtual Servers**, select a virtual server and, in the Action list, select Rename.

Managing Content Switching Policies

You can modify an existing policy by configuring rules or changing the URL of the policy, or you can remove a policy. You can also rename an existing advanced content switching policy. You can create different policies based on the URL. URL-based policies can be of different types, as described in the following table.

For more information, see Examples of URL-Based Policies pdf.

> **Note**
>
> You can configure rule-based content switching using classical policy expressions or advanced policy expressions.

To modify, remove, or rename a policy by using the command line interface

At the command prompt, type one of the following commands:

- `set cs policy <policyName> [-domain <domainValue>] [-rule <ruleValue>] [-url <URLValue>]`
- `rm cs policy <policyName>`
- `rename cs policy <policyName> <newPolicyName>`

Example:

```
set cs policy Policy-CS-1 -domain "www.domainxyz.com"

set cs policy Policy-CS-1 -rule "CLIENT.IP.SRC.SUBNET(22).EQ
    (10.100.148.0)"

set cs policy Policy-CS-2 -rule "SYS.TIME.BETWEEN(GMT 2010 Jun,GMT 2010
     Jul)"

set cs policy Policy-CS-1 -url /sports/*

rename cs policy Policy-CS-1 Policy-CS-11

rm cs policy Policy-CS-1
<!--NeedCopy-->
```

To modify, remove, or rename a policy by using the configuration utility

1. Navigate to **Traffic Management > Content Switching > Policies**.
2. Select the policy, and either delete it, edit it or, in the Action list, click **Rename**.

Managing client connections

September 21, 2020

To ensure efficient management of client connections, you can configure the content switching virtual servers on the NetScaler appliance.

- Configuring the ICMP Response. You can configure the NetScaler appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.
  The following settings can be made on a virtual server:
  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, the NetScaler appliance always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, the NetScaler appliance responds even if one virtual server is UP.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler appliance responds even if one virtual server set to ACTIVE is UP.

Redirecting Client Requests to a Cache

The NetScaler appliance cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the burden of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache and non-cacheable HTTP requests to the origin Web server. For more information on cache redirection, see "Cache Redirection."

To configure cache redirection on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -cacheable <Value>

Example

```
set cs vserver Vserver-CS-1 -cacheable yes
<!--NeedCopy-->
```

To configure cache redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Cacheable.

Enabling Delayed Cleanup of Virtual Server Connections

Under certain conditions, you can configure the down state flush setting to terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources and in certain cases speeds recovery of overloaded load balancing setups.

To configure the down state flush setting on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -downStateFlush <Value>

Example

```
set cs vserver Vserver-CS-1 -downStateFlush enabled
<!--NeedCopy-->
```

To configure the down state flush setting on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then select Down State Flush.

Rewriting Ports and Protocols for Redirection

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you may need to configure the NetScaler appliance to modify the port and the protocol to ensure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

To configure HTTP redirection on a virtual server by using the command line interface

At the command prompt, type:

set cs vserver <name> -redirectPortRewrite <Value>

Example

```
set cs vserver Vserver-CS-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

To configure HTTP redirection on a virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Rewrite.

Inserting the IP Address and Port of a Virtual Server in the Request Header

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, you must configure the required header tag on both virtual servers.

> **Note:** This option is not supported for wildcarded virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the command line interface

At the command prompt, type:

set cs vserver <name> -insertVserverIPPort <vServerIPPORT>

Example

```
set cs vserver Vserver-CS-1 -insertVserverIPPort 10.201.25.136:80
<!--NeedCopy-->
```

To insert the IP address and port of the virtual server in the client requests by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings and, in the Virtual Server IP Port Insertion list, select VIPADDR or V6TOV4MAPPING, and specify a port header in Virtual Server IP Port Insertion Value.

Setting a Time-out Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured time-out period elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection.

To set a time-out value for idle client connections by using the command line interface

At the command prompt, type:

set cs vserver <name> -cltTimeout <Value>

Example

```
set cs vserver Vserver-CS-1 -cltTimeout 100
<!--NeedCopy-->
```

To set a time-out value for idle client connections by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and specify a Client Idle Time-Out value.

Identifying Connections with the 4-tuple and Layer 2 Connection Parameters

You can now set the L2Conn option for a content switching virtual server. With the L2Conn option set, connections to the content switching virtual server are identified by the combination of the 4-tuple (<source IP>:<source port>::<destination IP>:<destination port>) and Layer 2 connection parameters. The Layer 2 connection parameters are the MAC address, VLAN ID, and channel ID.

To set the L2Conn option for a content switching virtual server by using the command line interface

At the command line, type the following commands to configure the L2Conn parameter for a content switching virtual server and verify the configuration:

set cs vserver <name> -l2Conn (**ON**	**OFF**)

-
- show cs vserver <name>

Example

```
> set cs vserver mycsvserver -l2Conn ON
Done
> show cs vserver mycsvserver
     mycsvserver (192.0.2.56:80) - HTTP   Type: CONTENT
     State: UP
         . . .
         . . .
     L2Conn: ON Case Sensitivity: ON
         . . .
         . . .
Done
>
<!--NeedCopy-->
```

To set the L2Conn option for a content switching virtual server by using the configuration utility

1. Navigate to Traffic Management > Content Switching > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and then select Layer 2 Parameters.

Troubleshooting

September 21, 2020

If the content switching feature does not work as expected after you have configured it, you can use some common tools to access NetScaler appliance resources and diagnose the problem.

Resources for Troubleshooting Content Switching

For best results, use the following resources to troubleshoot a content switching issue on a NetScaler appliance:

- Configuration file
- Relevant newnslog file
- Trace files
- Network topology diagram for the network setup of the customer
- Citrix documentation, such as release notes, Knowledge Center articles, and eDocs

In addition to the above resources, the following tools expedite troubleshooting:

- The iehttpheaders or a similar utility
- The Wireshark application customized for the NetScaler appliance trace files
- An SSH utility for command line access
- A HyperTerminal utility to access the console

Troubleshooting Content Switching Issues

The most common content switching issues involve the content switching feature not working at all, or working only intermittently, and Service Unavailable responses.

- **Issue**

  The content switching feature is not functioning.

  Resolution

  Check the configuration as follows:

  - Verify that the appliance is licensed for content switching.
  - Verify that the feature is enabled.
  - From the configuration file, verify that valid content switching policies are correctly bound to the load balancing virtual servers.

- **Issue**

  Client receives a 503 - Service Unavailable response.

  Resolution

  - Verify the URL and policy bindings. The client receives the 503 response when none of the policies you have configured is evaluated and no default load balancing virtual server is defined and bound to the content switching virtual server.
  - From the configuration, verify the policies and URL being accessed by the client.
  - Verify that for every type of request the respective policy is evaluated. If the policy is not evaluated, check the policy expression and update it if necessary.
  - Verify the URL and HTTP request and response headers. To do so, record an HTTPHeader trace and, if necessary, record the packet traces on the appliance and the client.

- **Issue**

  Intermittently, the content switching feature is not working as expected.

  Resolution

  - Study the network topology diagram, if available, of the setup to understand the various devices installed between the client and the server(s).

  - Verify the configuration and policy bindings. Make sure that the URL in the policy expression matches to the one in the client request.

- Verify that appropriate priorities are assigned to the policies. An incorrect precedence or priority assigned to a policy can cause a problem.
- Run the following commands to verify the bindings and the values of the policy hit counters in the output of the commands:

  show cs vserver <CS VServer>

  show cs policy <CS Policy>

  stat cs vserver <CS VServer>
- Using iehttpheaders or a similar utility, determine whether the HTTP headers for the requests or responses provide any pointers to the issue.
- Check the release notes and Knowledge Center articles.
- If the issue is still not resolved, contact Citrix Technical Support with appropriate data for further investigation.

DataStream

September 21, 2020

The NetScaler DataStream feature provides an intelligent mechanism for request switching at the database layer by distributing requests based on the SQL query being sent.

When deployed in front of database servers, a NetScaler appliance ensures optimal distribution of traffic from the application servers and Web servers. Administrators can segment traffic according to information in the SQL query and on the basis of database names, usernames, character sets, and packet size.

You can either configure load balancing to switch requests based on load balancing algorithms or elaborate the switching criteria by configuring content switching to make a decision based on an SQL query parameters. You can further configure monitors to track the state of database servers.

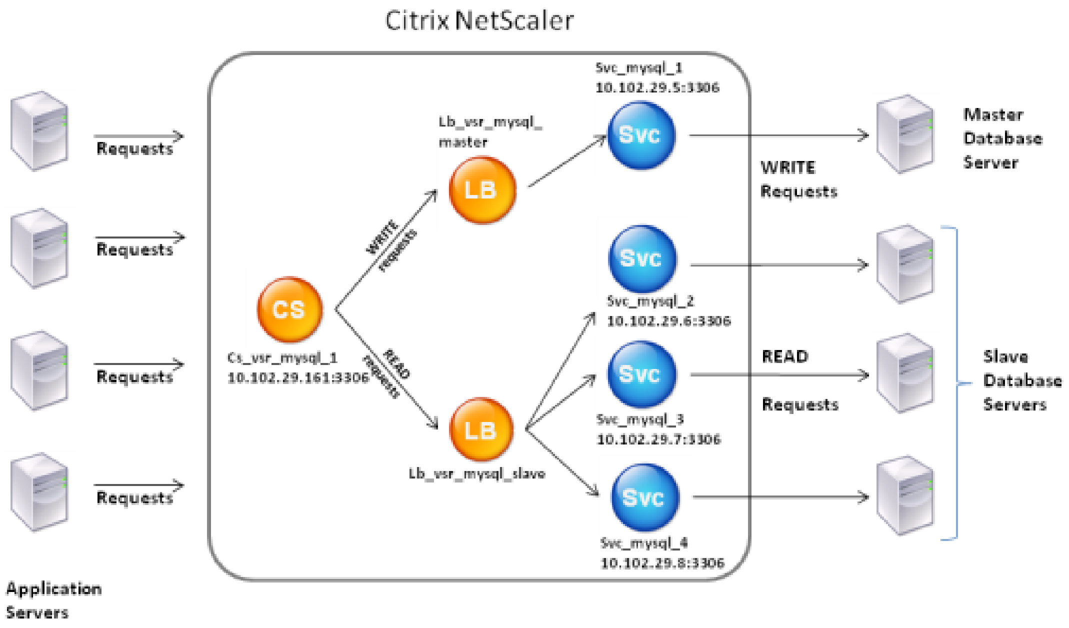
Note: NetScaler DataStream is supported only for MySQL and MS SQL databases. For information about the supported protocol version, character sets, special queries, and transactions, see DataStream Reference.

How DataStream Works

In DataStream, the ADC appliance is placed in-line between the application and/or Web servers and the database servers. On the appliance, the database servers are represented by services.

A typical DataStream deployment consists of the entities described in the following diagram.

Figure 1. DataStream Entity Model


Citrix NetScaler
Requests
Requests
Requests
Requests
Application Servers
CS
Cs_vsr_mysql_1
10.102.29.161:3306
WRITE requests
READ requests
LB
Lb_vsr_mysql_master
LB
Lb_vsr_mysql_slave
Svc
Svc_mysql_1
10.102.29.5:3306
Svc
Svc_mysql_2
10.102.29.6:3306
Svc
Svc_mysql_3
10.102.29.7:3306
Svc
Svc_mysql_4
10.102.29.8:3306
WRITE Requests
READ Requests
Master Database Server
Slave Database Servers

As shown in this figure, a DataStream configuration can consist of an optional content switching virtual server (CS), a load balancing setup consisting of load balancing virtual servers (LB1 and LB2) and services (Svc1, Svc2, Svc3, and Svc4), and content switching policies (optional).

The clients (application or Web servers) send requests to the IP address of a content switching virtual server (CS) configured on the NetScaler appliance. The appliance, then, authenticates the clients using the database user credentials configured on the appliance. The content switching virtual server (CS) applies the associated content switching policies to the requests. After evaluating the policies, the content switching virtual server (CS) routes the requests to the appropriate load balancing virtual server (LB1 or LB2), which, then, distributes the requests to the appropriate database servers (represented by services on the appliance) based on the load balancing algorithm. The NetScaler appliance uses the same database user credentials to authenticate the connection with the database server.

If a content switching virtual server is not configured on the appliance, the clients (application or Web servers) send their requests to the IP address of a load balancing virtual server configured on the appliance. The NetScaler appliance authenticates the client by using the database user credentials configured on the appliance, and then uses the same credentials to authenticate the connection with the database server. The load balancing virtual server distributes the requests to the database servers according to the load balancing algorithm. The most effective load balancing algorithm for database switching is the least connection method.

DataStream uses connection multiplexing to enable multiple client-side requests to be made over the same server-side connection. The following connection properties are considered:

- User name
- Database name
- Packet size
- Character set

Configuring Database Users

September 21, 2020

In databases, a connection is always stateful, which means that as soon as a connection is established, it must be authenticated.

You need to configure your database user name and password on the NetScaler appliance. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. When you add the database user names and passwords on the ADC, these are added to the nsconfig file.

> **Note**
>
> Names are case sensitive.

The ADC uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

To add a database user by using the command line interface

At the command prompt, type

```
add db user <username> - password <password>
<!--NeedCopy-->
```

Example:

```
add db user nsdbuser -password dd260427edf
<!--NeedCopy-->
```

To add a database user by using the configuration utility

Navigate to **System > User Administration > Database Users**, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the ADC appliance.

To reset the password of a database user by using the command line interface

At the command prompt, type

```
set db user <username> -password <password>
<!--NeedCopy-->
```

Example:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

To reset the password of database users by using the configuration utility

Navigate to **System > User Administration > Database Users**, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the ADC appliance. However, if the user continues to exist on the database server and you remove the user from the ADC appliance, any request from the client with this user name does not get authenticated, and therefore, does not get routed to the database server.

To remove a database user by using the command line interface

At the command prompt, type

```
rm db user <username>
<!--NeedCopy-->
```

Example:

```
rm db user nsdbuser
<!--NeedCopy-->
```

To remove a database user by using the configuration utility

Navigate to **System > User Administration > Database Users**, select a user, and click **Delete**.

Configuring a Database Profile

September 21, 2020

A database profile is a named collection of parameters that is configured once but applied to multiple virtual servers that require those particular parameter settings. After creating a database profile, you bind it to load balancing or content switching virtual servers. You can create as many profiles as you need.

To create a database profile by using the command line interface

At the command line, type the following commands to create a database profile and verify the configuration:

```
add db dbProfile <name> [-interpretQuery ( YES | NO )] [-stickiness (
    YES | NO )] [-kcdAccount <string>]

show db dbProfile
<!--NeedCopy-->
```

Example:

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -
    kcdAccount mykcdaccnt
 Done
> show dbProfile myDBProfile
     Name: myDBProfile
     Interpret Query: YES
     Stickyness: YES
     KCD Account: mykcdaccnt
     Reference count: 0

 Done
>
<!--NeedCopy-->
```

To create a database profile by using the configuration utility

Navigate to **System > Profiles** and, on the **Database Profiles** tab, configure a database profile.

To bind a database profile to a load balancing or content switching virtual server by using the command line interface

At the command line, type:

```
set (lb | cs) vserver <name> -dbProfileName <string>
<!--NeedCopy-->
```

To bind a database profile to a load balancing or content switching virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers or Traffic Management > Content Switching > Virtual Servers**, and open a virtual server.
2. In **Advanced Settings**, select **Profiles** and, in the **DB Profile** list, select a profile to bind to the virtual server. To create a new profile, click plus (+).

Configuring Load Balancing for DataStream

September 21, 2020

Before configuring a load balancing setup, you must enable the load balancing feature. Then, begin by creating at least one service for each database server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server and bind the services to the virtual server.

> **Note**:
>
> For databases, the load balancing can only occur on homogenous database servers (database servers that contain the exact same databases). For a configuration that contains unique databases on different servers, you must use content switching. If some of your database servers host identical content, you can use load balancing on those servers only. You can then use content switching policies to send requests to the load balancing virtual server that manages load balancing for those databases.
>
> The NetScaler appliance currently stores the database name and login information during the database session. When a query is made to the database, it uses that information to connect to the specific database server.

Parameter values specific to DataStream

- Protocol

Use the
MYSQL protocol type for MySQL databases and
MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see
http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see
http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

- Port

  Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

- Method

  It is recommended that you use the Least Connection method for better load balancing and lower server load. However, other methods, such as Round Robin, Least Response Time, Source IP Hash, Source IP Destination IP Hash, Least Bandwidth, Least Packets, and Source IP Source Port Hash, are also supported.
  Note: URL Hash method is not supported for DataStream.

- MS SQL Server Version

  If you are using the Microsoft SQL Server, and you expect some clients to not be running the same version as your Microsoft SQL Server product, set the
  Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the
  Server Version parameter, see
  Configuring the MySQL and Microsoft SQL server version setting.

- MySQL Server Version

  If you are using the MySQL Server, and you expect some clients to not be running the same version as your MySQL Server product, set the Server Version parameter for the load balancing virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the
  Server Version parameter, see
  Configuring the MySQL and Microsoft SQL server version setting.

Configuring Content Switching for DataStream

September 21, 2020

You can segment traffic according to information in the SQL query, on the basis of database names, user names, character sets, and packet size.

You can configure content switching policies with default syntax expressions to switch content based on connection properties, such as user name and database name, command parameters, and the SQL query to select the server.

The default syntax expressions evaluate traffic associated with MYSQL and MS SQL database servers. You can use request-based expressions in default syntax policies to make request switching decisions at the content switching virtual server bind point and response-based expressions (expressions that begin with MYSQL.RES) to evaluate server responses to user-configured health monitors.

For information about default syntax expressions, see
Default Syntax Expressions: DataStream.

> **Note**:
>
> For databases, the load balancing can only occur on homogenous database servers (database servers that contain the exact same databases). For a configuration that contains unique databases on different servers, you must use content switching. If some of your database servers host identical content, you can use load balancing on those servers only. You can then use content switching policies to send requests to the load balancing virtual server that manages load balancing for those databases.
>
> The NetScaler appliance currently stores the database name and login information during the database session. When a query is made to the database, it uses that information to connect to the specific database server.

Parameter values specific to DataStream

- Protocol

  Use the
  MYSQL protocol type for MySQL databases and
  MSSQL protocol type for MS SQL databases while configuring virtual servers and services. The MySQL and TDS protocols are used by the clients to communicate with the respective database servers by using SQL queries. For information about the MySQL protocol, see
  http://dev.mysql.com/doc/internals/en/client-server-protocol.html. For information about the TDS protocol, see
  http://msdn.microsoft.com/en-us/library/dd304523(v=prot.13).aspx.

- Port

  Port on which the virtual server listens for client connections. Use port 3306 for MySQL database servers.

- MS SQL Server Version

  If you are using Microsoft SQL Server, and you expect some clients to not be running the same version as your Microsoft SQL Server product, set the
  Server Version parameter for the content switching virtual server. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version. For more information about setting the
  Server Version parameter, see
  Configuring the Microsoft SQL Server Version Setting.

Configuring Monitors for DataStream

September 21, 2020

To track the state of each load balanced database server in real time, you need to bind a monitor to each service. The monitor is configured to test the service by sending periodic probes to the service. (This is sometimes referred to as performing a health check.) If the monitor receives a timely response to its probes, it marks the service as UP. If it does not receive a timely response to the designated number of probes, it marks the service as DOWN.

For DataStream, you need to use the built-in monitors, MYSQL-ECV and MSSQL-ECV. This monitor provides the ability to send an SQL request and parse the response for a string.

Before configuring monitors for DataStream, you must add database user credentials to your NetScaler appliance. For information about configuring monitors, see Configure monitors in a load balancing setup.

When you create a monitor, a TCP connection is established with the database server, and the connection is authenticated by using the user name provided while creating the monitor. You can then run an SQL query to the database server and evaluate the server response to check whether it matches the configured rule.

The following examples are for MYSQL servers.

Examples:

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MYSQL-ECV -sqlQuery "select * from
```

```
table2;" -evalrule "mysql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MYSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mysql.res.atleast_rows_count(7)" -database "NS" -userName "user2"
<!--NeedCopy-->
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MYSQL-ECV
-sqlQuery "select * from ABC;" -evalrule "mysql.res.row(1).double_elem(2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

The following examples are for MSSQL servers.

Examples:

In the following example, the value of the error message is evaluated to determine the state of the server.

```
add lb monitor lb_mon1 MSSQL-ECV -sqlQuery "select * from
table2;" -evalrule "mssql.res.error.message.contains("Invalid
User")"-database "NS" -userName "user1"
<!--NeedCopy-->
```

In the following example, the number of rows in the response is evaluated to determine the state of the server.

```
add lb monitor lb_mon4 MSSQL-ECV -sqlQuery "select * from
table4;" -evalrule "mssql.res.atleast_rows_count(7)" -database "NS" -userName "user2"
<!--NeedCopy-->
```

In the following example, the value of a particular column is evaluated to determine the state of the server.

```
add lb monitor lb_mon3 MSSQL-ECV
```

```
-sqlQuery "select * from ABC;" -evalrule "mssql.res.row(1).double_elem(2) == 345.12"
-database "NS" -userName "user3"
<!--NeedCopy-->
```

Use Case 1: Configuring DataStream for a Master/Slave Database Architecture

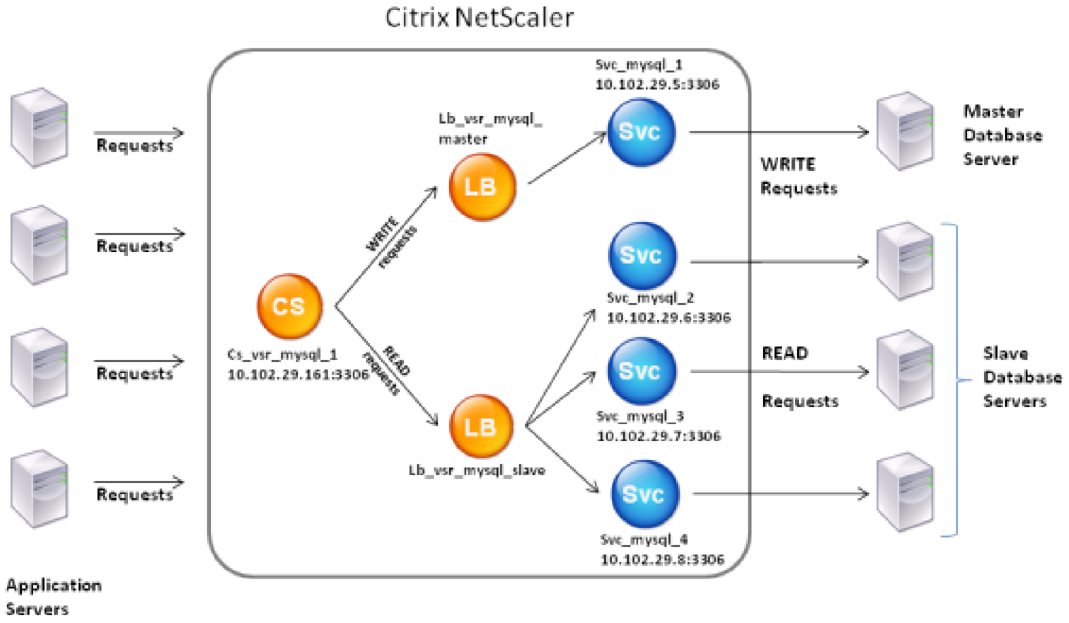
September 21, 2020

A commonly used deployment scenario is the master/slave database architecture where the master database replicates all information to the slave databases.

For master/slave database architecture, you may want all WRITE requests to be sent to the master database and all READ requests to the slave databases.

The following figure shows the entities and the values of the parameters you need to configure on the appliance.

Figure 1. DataStream Entity Model for Master/Slave Database Setup


Citrix NetScaler
Requests
Requests
Requests
Requests
Application Servers
CS
Cs_vsr_mysql_1
10.102.29.161:3306
WRITE requests
READ requests
LB
Lb_vsr_mysql_master
LB
Lb_vsr_mysql_slave
Svc
Svc_mysql_1
10.102.29.5:3306
Svc
Svc_mysql_2
10.102.29.6:3306
Svc
Svc_mysql_3
10.102.29.7:3306
Svc
Svc_mysql_4
10.102.29.8:3306
WRITE Requests
READ Requests
Master Database Server
Slave Database Servers

In this example scenario, a service (Svc_mysql_1) is created to represent the master database and is bound to a load balancing virtual server (Lb_vsr_mysql_master). Three more services (Svc_mysql_2, Svc_mysql_3, and Svc_mysql_4) are created to represent the three slave databases, and they are bound to another load balancing virtual server (Lb_vsr_mysql_slave).

A content switching virtual server (Cs_vsr_mysql_1) is configured with associated policies to send all WRITE requests to the load balancing virtual server, Lb_vsr_mysql_master, and all READ requests to the load balancing virtual server, Lb_vsr_mysql_slave.

When a request reaches the content switching virtual server, the virtual server applies the associated content switching policies to that request. After evaluating the policies, the content switching virtual server routes the request to the appropriate load balancing virtual server, which sends it to the appropriate service.

The following table lists the names and values of the entities and the policy configured on the NetScaler appliance.

Entity Type	Name	IP Address	Protocol	Port	Expression
Services	Svc_mysql_1	10.102.29.5	MYSQL	3306	NA

Entity Type	Name	IP Address	Protocol	Port	Expression
	Svc_mysql_2	10.102.29.6	MYSQL	3306	NA
	Svc_mysql_3	10.102.29.7	MYSQL	3306	NA
	Svc_mysql_4	10.102.29.8	MYSQL	3306	NA
Load balancing virtual servers	Lb_vsr_mysql_	10.102.29.201	MYSQL	3306	NA
	Lb_vsr_mysql_slave	10.102.29.202	MYSQL	3306	NA
Content switching virtual server	Cs_vsr_mysql_	10.102.29.161	MYSQL	3306	NA
Content switching policy	Cs_select	NA	NA	NA	“MYSQL.REQ.QUERY.COMMA

Table 1. Entity and Policy Names and Values

To configure DataStream for a master/slave database setup by using the command line interface

At the command prompt, type

```
add service Svc_mysql_1 10.102.29.5 mysql 3306

add service Svc_mysql_2 10.102.29.6 mysql 3306

add service Svc_mysql_3 10.102.29.7 mysql 3306

add service Svc_mysql_4 10.102.29.8 mysql 3306

add lb vserver Lb_vsr_mysql_master mysql 10.102.29.201 3306

add lb vserver Lb_vsr_mysql_slave mysql 10.102.29.202 3306

bind lb vserver Lb_vsr_mysql_master svc_mysql_1

bind lb vserver Lb_vsr_mysql_slave svc_mysql_2

bind lb vserver Lb_vsr_mysql_slave svc_mysql_3

bind lb vserver Lb_vsr_mysql_slave svc_mysql_4

add cs vserver Cs_vsr_mysql_1 mysql 10.102.29.161 3306

add cs policy Cs_select -rule "MYSQL.REQ.QUERY.COMMAND.contains("select")"

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_master

bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_slave -policy Cs_select -priority 10
``
<!--NeedCopy-->
```

Use Case 2: Configuring the Token Method of Load Balancing for DataStream

September 21, 2020

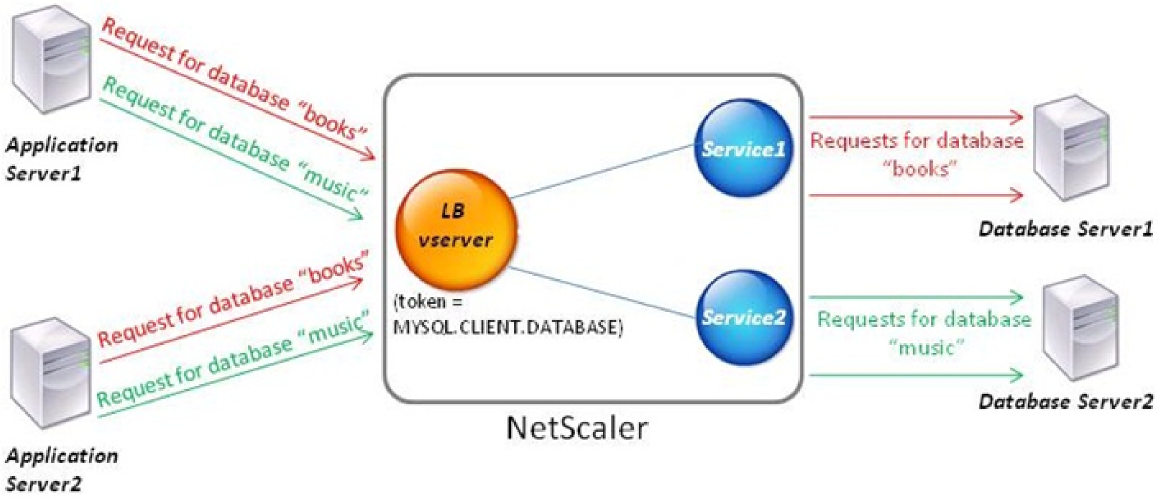
You can configure the token method of load balancing for DataStream to base the selection of database servers on the value of the token extracted from the client (application or web server) requests. These tokens are defined by using SQL expressions. For subsequent requests with the same token, the NetScaler appliance sends the requests to the same database server that handled the initial request. Requests with the same token are sent to the same database server until the maximum connection limit is reached or the session entry has aged out.

You can use the following sample SQL expressions to define tokens:

MySQL	MS SQL
MYSQL.REQ.QUERY.TEXT	MSSQL.REQ.QUERY.TEXT
MYSQL.REQ.QUERY.TEXT(n)	MSSQL. REQ.QUERY.TEXT(n)
MYSQL.REQ.QUERY.COMMAND	MSSQL.REQ.QUERY.COMMAND
MYSQL.CLIENT.USER	MSSQL.CLIENT.USER
MYSQL.CLIENT.DATABASE	MSSQL.CLIENT.DATABASE
MYSQL.CLIENT.CAPABILITIES	

The following example shows how the NetScaler DataStream feature works when you configure the token method of load balancing.

Figure 1. How DataStream Works with the Token Method of Load Balancing


Application Server1
Request for database "books"
Request for database "music"
Application Server2
Request for database "books"
Request for database "music"
LB vserver
(token = MYSQL.CLIENT.DATABASE)
Service1
Service2
NetScaler
Requests for database "books"
Database Server1
Requests for database "music"
Database Server2

In this example, the token is the name of the database. A request with token books is sent to Database Server1 and a request with token music is sent to Database Server2. All subsequent requests with token books are sent to Database Server1 and requests with token music are sent to Database Server2. This configuration provides pseudo persistence with the database servers.

To configure this example by using the command line interface

At the command prompt, type:

```
add service Service1 192.0.2.9 MYSQL 3306

add service Service2 192.0.2.11 MYSQL 3306

add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -rule MYSQL.CLIENT.DATABASE

bind lb vserver token_lb_vserver Service1

bind lb vserver token_lb_vserver Service2
<!--NeedCopy-->
```

To configure this example by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, configure a virtual server and specify the protocol as MYSQL.
2. Click in the **Service** section, and configure two services specifying the protocol as MYSQL. Bind these services to the virtual server.
3. In **Advanced Settings**, click **Method** and, in the **Load Balancing Method** list, select TOKEN and specify the expression as MYSQL.CLIENT.DATABASE.

Use Case 3: Logging MSSQL Transactions in Transparent Mode

September 21, 2020

You can configure the NetScaler appliance to operate transparently between MSSQL clients and servers, and to only log or analyze details of all client-server transactions. Transparent mode is designed so that the NetScaler appliance only forwards MSSQL requests to the server, and then relays the server's responses to the clients. As the requests and responses pass through the appliance, the appliance logs information gathered from them, as specified by the audit logging or AppFlow configuration, or collects statistics, as specified by the Action Analytics configuration. You do not have to add database users to the appliance.

When operating in transparent mode, the NetScaler appliance does not perform load balancing, content switching, or connection multiplexing for the requests. However, it responds to a client's pre-login packet on behalf of the server so that it can prevent encryption from being agreed upon during the pre-login handshake. The login packet and subsequent packets are forwarded to the server.

Summary of Configuration Tasks

For logging or analyzing MSSQL requests in transparent mode, you have to do the following:

- Configure the NetScaler appliance as the default gateway for both clients and servers.
- Do one of the following on the NetScaler appliance:
  - **If you can configure the use source IP address (USIP) option globally,** create a load balancing virtual server with a wildcard IP address and the port number on which the MSSQL servers listen for requests (a port-specific wildcard virtual server). Then, enable the USIP option globally. If you configure a port-specific wildcard virtual server, you do not have to create MSSQL services on the appliance. The appliance discovers the services on the basis of the destination IP address in the client requests.
  - **If you do not want to configure the USIP option globally,** create MSSQL services with the USIP option enabled on each of them. If you configure services, you do not have to create a port-specific wildcard virtual server.

- Configure audit logging, AppFlow, or Action Analytics to log or collect statistics about the requests. If you configure a virtual server, you can bind your policies either to the virtual server or to the global bind point. If you do not configure a virtual server, you can bind your policies to only the global bind point.

Configuring Transparent Mode by Using a Wildcard Virtual Server

You can configure transparent mode by configuring a port-specific wildcard virtual server and enabling Use Source IP (USIP) mode globally. When a client sends its default gateway (the NetScaler appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance checks whether the destination IP address is available. If the IP address is available, the virtual server forwards the request to the server. Otherwise, it drops the request.

To create a wildcard virtual server by using the command line

At the command prompt, type the following commands to create a wildcard virtual server and verify the configuration:

```
add lb vserver <name> <serviceType> <IPAddress> <port>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
> add lb vserver wildcardLbVs MSSQL * 1433
 Done
> show lb vserver wildcardLbVs
    wildcardLbVs (*:1433) - MSSQL   Type: ADDRESS
    State: UP
    . . .

 Done
>
<!--NeedCopy-->
```

To create a wildcard virtual server by using the GUI

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and create a virtual server. Specify MSSQL as the protocol and * as the IP address.

To enable Use Source IP (USIP) mode globally by using the CLI

At the command prompt, type the following commands to enable USIP mode globally and verify the configuration:

```
enable ns mode USIP

show ns mode
<!--NeedCopy-->
```

Example:

```
> enable ns mode USIP
 Done
> show ns mode

    Mode                              Acronym
       Status
    -------                           -------
       ------
               . . .
 3) Use Source IP                  USIP                          ON
               . . .
 Done
>
<!--NeedCopy-->
```

To enable USIP mode globally by using the GUI

1. Navigate to **System > Settings** and, in Modes and Features, select Configure Modes.
2. Select Use Source IP.

Configuring Transparent Mode by Using MSSQL Services

You can configure transparent mode by configuring MSSQL services and enabling USIP on each service. When a client sends its default gateway (the NetScaler appliance) a request with the IP address of an MSSQL server in the destination IP address header, the appliance forwards the request to the destination server.

To create an MSSQL service and enable USIP mode on the service by using the command line interface

At the command prompt, type the following commands to create an MSSQL service, with USIP enabled, and verify the configuration:

```
add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES

show service <name>
<!--NeedCopy-->
```

Example:

```
> add service myDBservice 192.0.2.0 MSSQL 1433 -usip YES
 Done
> show service myDBservice
    myDBservice (192.0.2.0:1433) - MSSQL
    State: UP
                   . . .
    Use Source IP: YES      Use Proxy Port: YES
                   . . .
 Done
>
<!--NeedCopy-->
```

To create an MSSQL service, with USIP enabled, by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and configure a service.
2. Specify protocol as MSSQL and, in Settings, select Use Source IP.

Use Case 4: Database Specific Load Balancing

September 21, 2020

A database server farm must be load balanced not only based on the states of the servers, but also based on the availability of the database on each server. A service might be up, and a load balancing device might show it as being in the UP state, but the requested database might be unavailable on that service. The request is not served if a query is forwarded to a service on which the database in unavailable. Therefore, a load balancing device must be aware of the availability of a database on each service. And when making a load balancing decision, it must consider only those services on which the database is available.

As an example, consider that database servers server1, server2, and server3 host databases mydatabase1 and mydatabase2. If mydatabase1 becomes unavailable on server2, the load balancing device must be aware of that change in state. It must load balance requests for mydatabase1 across only server1 and server3. After mydatabase1 becomes available on server2, the load balancing device must include server2 in load balancing decisions. Similarly, if mydatabase2 becomes unavailable on server3, the device must load balance requests for mydatabase2 across only server1 and server2. It must include server3 in its load balancing decisions only when mydatabase2 becomes available. This load balancing behavior must be consistent across all the databases that are hosted on the server farm.

The NetScaler appliance implements this behavior by retrieving a list of all the databases that are active on a service. To retrieve the list of active databases, the appliance uses a monitor that is configured with an appropriate SQL query. If the requested database is unavailable on a service, the appliance excludes the service from load balancing decisions until it becomes available. This behavior ensures uninterrupted service to clients.

> **Note**
>
> Database specific load balancing is supported for only MSSQL and MySQL service types. This support is also available for Microsoft SQL Server 2012 high availability deployment.

To set up database specific load balancing, you must configure the following:

- Enable the load balancing feature, and configure a load balancing virtual server of type MSSQL or MySQL.
- Configure the services that host the database, and bind the services to the virtual server. The monitor needs valid user credentials to log on to the database server, so you must configure a database user account on each of the servers and then add the user account to the NetScaler appliance.
- Then, you configure an MSSQL-ECV or MYSQL-ECV monitor and bind the monitor to each service.
- Finally, you must test the configuration to ensure that it is working as intended. Before you perform these configuration tasks, make sure you understand how database specific load balancing works.

How Database Specific Load Balancing Works

For database specific load balancing, you configure a monitor that periodically queries each database server for the names of all the active databases on it. The NetScaler appliance stores the results, and regularly updates the records based on the information retrieved through monitoring. When a client queries a particular database, the appliance uses the configured load balancing method to select a service. It then checks the records to determine whether the database is available on that service. If the records indicate that the database is not available, it uses the configured load balancing method to select the next available service, and then repeats the check. The appliance forwards the query to the first available service on which the database is active.

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled. The entities does not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB

show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   OFF
 2)     Surge Protection               SP                   ON
 3)     Load Balancing                 LB                   ON
 .
 .
 .
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

To enable load balancing by using the configuration utility

Navigate to **System > Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configuring a Load Balancing Virtual Server for Database Specific Load Balancing

To configure a virtual server to load balance databases based on the availability, you enable the database specific load balancing parameter on the virtual server. Enabling the parameter modifies the load balancing logic that the NetScaler appliance refers the results of the monitoring probe sent to the selected service, before forwarding the query to that service.

> **Note**
>
> For configuration examples related to MSSQL or MySQL, refer to the following topics:
>
> Configuration examples for MSSQL virtual server and Configuration examples for MySQL virtual server.

To configure a load balancing virtual server for database specific load balancing

At the command prompt, type the following command to configure a load balancing virtual server for database specific load balancing and verify the configuration:

```
add lb vserver <name> <serviceType> <ipAddress> <port> -dbsLb ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the NetScaler appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

Configuring Database Users

In databases, a connection is always stateful, which means that when a connection is established, it must be authenticated.

You configure your database user name and password on the NetScaler. For example, if you have a user John configured on the database, you need to configure the user John on the ADC too. When you add the database user names and passwords on the NetScaler appliance, the same are added to the nsconfig file.

> **Note**
>
> Names are case sensitive.

The NetScaler uses these user credentials to authenticate the clients, and then authenticate the server connections with the database servers.

To add a database user by using the command line interface

At the command prompt, type

```
add db user <username> - password <password>
<!--NeedCopy-->
```

Example:

```
add db user nsdbuser -password dd260427edf
<!--NeedCopy-->
```

To add a database user by using the configuration utility

Navigate to **System > User Administration > Database Users**, and configure a database user.

If you have changed the password of the database user on the database server, you must reset the password of the corresponding user configured on the NetScaler appliance.

To reset the password of a database user by using the command line interface

At the command prompt, type

```
set db user <username> -password <password>
<!--NeedCopy-->
```

Example:

```
set db user nsdbuser -password dd260538abs
<!--NeedCopy-->
```

To reset the password of database users by using the configuration utility

Navigate to **System > User Administration > Database Users**, select a user, and enter new values for the password.

If a database user no longer exists on the database server, you can remove the user from the NetScaler appliance. However, if the user continues to exist on the database server and you remove the user from the NetScaler appliance, any request from the client with this user name does not get authenticated. Therefore, the user name does not get routed to the database server.

To remove a database user by using the command line interface

At the command prompt, type

```
rm db user <username>
<!--NeedCopy-->
```

Example:

```
rm db user nsdbuser
<!--NeedCopy-->
```

To remove a database user by using the configuration utility

Navigate to **System > User Administration > Database Users**, select a user, and click **Delete**.

Configuring a Monitor to Retrieve the Names of Active Databases

You can create a monitor to retrieve the list of all active databases on a database instance. The monitor logs on to the database server by using a valid user credentials and runs an appropriate SQL query. The SQL query you need to use depends on your SQL server deployment. For example, in an MSSQL database mirroring setup, you can use the following query to retrieve a list of active databases available on a server instance.

```
select name from sys.databases where state=0
<!--NeedCopy-->
```

In a MySQL database setup you can use the following queries to retrieve a list of active databases available on a server instance.

Show databases:

You also configure the monitor to evaluate the response for an error condition, and to store the results if there is no error. If the response contains an error, the monitor marks the service as DOWN. The appliance excludes the service from load balancing decisions until an error is no longer returned.

> **Note**
>
> The database specific load balancing feature is supported only for the MSSQL and MySQL service types. Therefore, the monitor type must be MSSQL-ECV or MYSQL-ECV.

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line

At the command prompt, type the following commands to retrieve the names of all the active databases hosted on a service and verify the configuration:

```
add lb monitor <monitorName> <type> -userName <string> -sqlQuery <text>
    -evalRule <expression> -storedb ENABLED

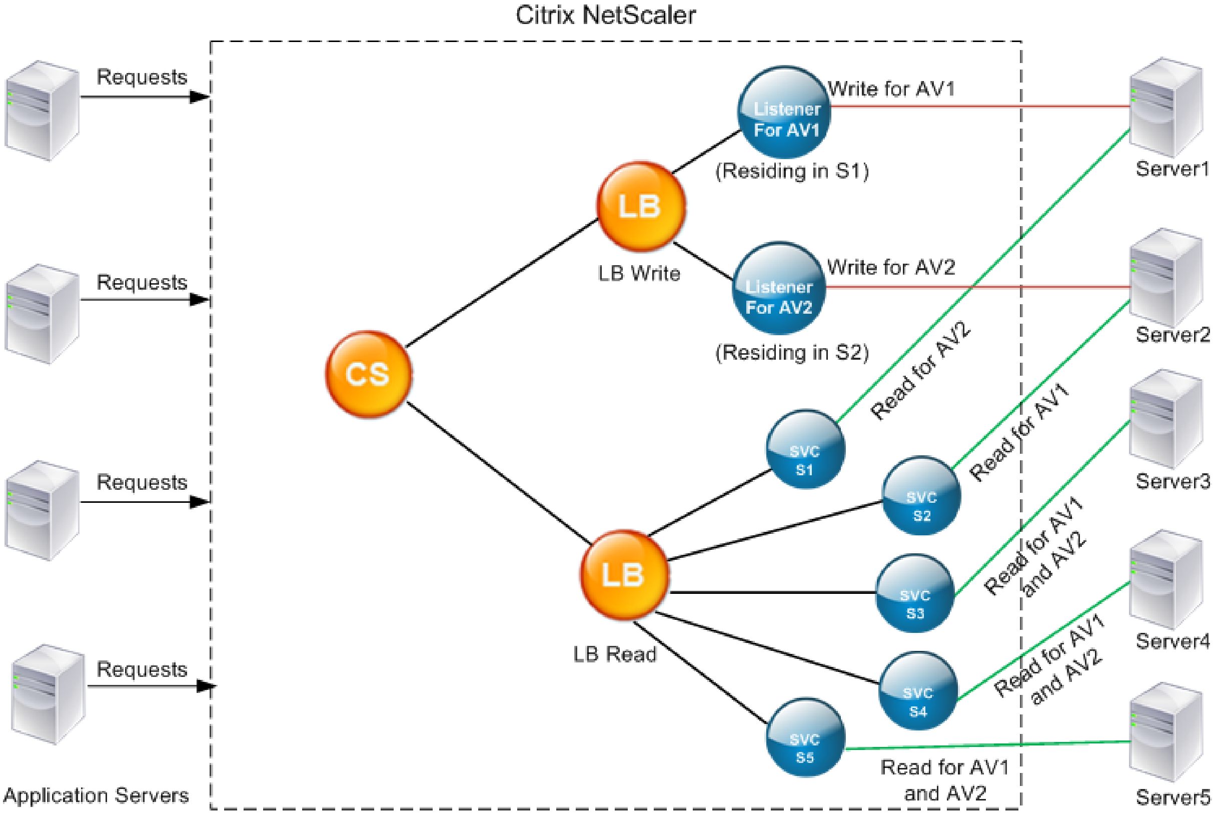
show lb monitor <monitorName>
<!--NeedCopy-->
```

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors** and configure a monitor of type MSSQL-ECV or MYSQL-ECV.
2. In **Special Parameters**, specify a user name, query, and a rule. For example, for MSSQL-ECV, the query must be "select name from sys.databases where state=0"), and a rule must be MSSQL.RES.TYPE.NE(ERROR). For MYSQL-ECV, the query must be "show databases" and a rule must be MYSQL.RES.TYPE.NE(ERROR).

Availability Groups Deployment Support for MSSQL

Consider the following scenario in which database specific load balancing is configured in a high availability group deployment. S1 through S5 is the services on the ADC appliance. DB1 through DB4 is the databases on the servers represented by the services S1 through S5. AV1 and AV2 are the availability groups. Each availability group contains up to one primary database server instance and up to four secondary database server instances. A service, representing the servers in the availability group, can be primary for one availability group and secondary for another availability group. Each availability group contains different databases and one listener, which is a service. All requests arrive on the listener service that resides on the primary database. AVI contains databases DB1 and DB2. AV2 contains databases DB3 and DB4. L1 and L2 are the listeners on AV1 and AV2 respectively. S1 is the primary service for AV1 and S2 is the primary service for AV2.


Citrix NetScaler
Requests
Requests
Requests
Requests
Application Servers
CS
LB
LB Write
LB
LB Read
Listener For AV1
(Residing in S1)
Listener For AV2
(Residing in S2)
Write for AV1
Write for AV2
SVC S1
SVC S2
SVC S3
SVC S4
SVC S5
Read for AV2
Read for AV1
Read for AV1 and AV2
Read for AV1 and AV2
Read for AV1 and AV2
Server1
Server2
Server3
Server4
Server5

Service	List of Active Databases on the Service
S1	DB1, DB2, DB3, DB4
S2	DB3, DB4
S3	DB3, DB4
S4	DB1, DB2
S5	DB1, DB2

Availability Group	Databases	Services representing the Servers in Availability Group
AV1	DB1, DB2	S1, S4, S5
AV2	DB3, DB4	S1, S2, S3

Queries flow as follows:

1. A READ query for AV1 is load balanced between S4 and S5. S1 is the primary for AV1.
2. A WRITE query for AV1 is directed to L1.
3. A READ query for AV2 is load balanced between S1 and S3. S2 is the primary for AV2.
4. A WRITE query for AV1 is directed to L2.

Sample Configuration

1. Configure load balancing and content switching virtual servers.
    - `add lb vserver lbwrite -dbslb enabled`
    - `add lbvserver lbread MSSQL -dbslb enabled`
    - `add csvserver csv MSSQL 1.1.1.10 1433`
2. Configure two listener services, one for each availability group, and five services S1 through S5 representing databases DB1 through DB4.
    - `add service L1 1.1.1.11 MSSQL 1433`
    - `add service L2 1.1.1.12 MSSQL 1433`
    - `add service s1 1.1.1.13 MSSQL 1433`
    - `add service s2 1.1.1.14 MSSQL 1433`
    - `add service s3 1.1.1.15 MSSQL 1433`
    - `add service s4 1.1.1.16 MSSQL 1433`
    - `add service s5 1.1.1.17 MSSQL 1433`
3. Bind the services to the load balancing virtual servers.
    - `bind lbvserver lbwrite L1`
    - `bind lbvserver lbwrite L2`
    - `bind lbvserver lbread s1`
    - `bind lbvserver lbread s2`
    - `bind lbvserver lbread s3`
    - `bind lbvserver lbread s4`
    - `bind lbvserver lbread s5`
4. Configure database users.
    - `add db user nsdbuser1 -password dd260427edf`
    - `add db user nsdbuser2 -password ccd1234xyzw`
5. Configure two monitors, monitor_L1 and monitor_L2 for each listener service, to retrieve the list of active databases in that availability group. Add a monitor, monitor1 to retrieve the list of databases for the secondary database server instance.
    - `add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery " SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_ b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.11'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)” -storedb`

  `ENABLED`

  - `add lb monitor monitor_L2 MSSQL-ECV -userNameuser1 -sqlQuery " SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replicca b ON a.replica_id=b.replica_id INNER JOIN sys.availability_group_listeners c on b.group_id = c.group_id INNER JOIN sys.availability_group_listener_ip_a d on c.listener_id = d.listener_id WHERE b.role = 1 and d.ip_address like '1.1.1.12'"-evalRule "MSSQL.RES.TYPE.NE(ERROR)"-storedb ENABLED`
  - `add lb monitor monitor1 MSSQL-ECV -userNameuser1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states b ON a.replica_id=b.replica_id WHERE b.role = 2"-evalRule "MSSQL.RES.TYPE.NE(ERROR)"-storedb ENABLED`

6. Configure read and write policies.
    - `add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(" insert")"`
    - `add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS("select ")"`
7. Bind the policies to the content switching virtual server.
    - `bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11`
    - `bind csvserver csv -targetLBVserver lbread -policyName pol_read -priority 12`
8. Bind monitors to the services. Bind monitors to services L1 and L2 to get the list of active databases for the availability group for which it is the listener. Bind monitors to all the services that are bound to the read-only virtual server.
    - `bind service L1 -monitorName monitor_L1`
    - `bind service L2 -monitorName monitor_L2`
    - `bind service s1 -monitorName monitor1`
    - `bind service s2 -monitorName monitor1`
    - `bind service s3 -monitorName monitor1`
    - `bind service s4 -monitorName monitor1`
    - `bind service s5 -monitorName monitor1`

Configuration examples for MSSQL virtual server

To configure a load balancing virtual server for database specific load balancing:

```
add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED

 Done
```

```

<!--NeedCopy-->
```

```
    show lb vserver DBSpecificLB1
    DBSpecificLB1 (192.0.2.10:1433) - MSSQL Type ADDRESS
    . . .
    DBS_LB: ENABLED

    Done
<!--NeedCopy-->
```

To configure services:

```
    add service msservice1 5.5.5.5 MSSQL 1433
<!--NeedCopy-->
```

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line:

```
    add lb monitor mssql-monitor1 MSSQL-ECV -userName user1 -sqlQuery "
       select name from sys.databases where state=0" -evalRule "MSSQL.
       RES.TYPE.NE(ERROR)" -storedb EN

    Done
<!--NeedCopy-->
```

```
    show lb monitor mssql-monitor1

    1)   `Name.......: mssql-monitor1    Type......: MSSQL-ECV`

    ...

    Special parameters: Database.....:""

    User name.....:"user1"

    Query..:select name from sys.databases where state=0 EvalRule...:
       MSSQL.RES.TYPE.NE(ERROR)

    Version...:70 STORE_DB...:ENABLED

    Done
<!--NeedCopy-->
```

Configuration examples for MySQL virtual server

To configure a load balancing virtual server for database specific load balancing:

```
    add lb vserver DBSpecificLB1 MYSQL 192.0.2.10 3306 -dbsLb ENABLED

    Done
<!--NeedCopy-->
```

```
    show lb vserver DBSpecificLB1

    DBSpecificLB1 (192.0.2.10:3306) - MYSQL Type: ADDRESS

    . . .

    DBS_LB: ENABLED

    Done
<!--NeedCopy-->
```

To configure services:

`add service msservice1 5.5.5.5 MYSQL 3306`

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the command line:

```
    add lb monitor mysql-monitor1 MYSQL-ECV -userName user1 -sqlQuery "
       show databases" -evalRule "MYSQL.RES.TYPE.NE(ERROR)" -storedb
       ENABLED

    Done

<!--NeedCopy-->
```

```
    show lb monitor mysql-monitor1

    1)      Name.......: mysql-monitor1  Type......: MYSQL-ECV  State
       ....:  ENABLED

    ...

    Special parameters: Database.....:""

    User name.....:"user1" Query..:show databases

    EvalRule...:MYSQL.RES.TYPE.NE(ERROR) STORE_DB...:ENABLED

    Done
<!--NeedCopy-->
```

DataStream Reference

September 21, 2020

This reference describes the MySQL and TDS protocols, the database versions, the authentication methods, and the character sets supported by the DataStream feature. It also describes how the NetScaler handles transaction requests and special queries that modify the state of a connection.

You can also configure the NetScaler appliance to generate audit log messages for the DataStream feature.

Supported Database Versions, Protocols, and Authentication Methods

	MySQL Database	MS SQL Database
Database Versions	MySQL database versions 4.1, 5.0, 5.1, 5.4, 5.5, and 5.6.	MS SQL database versions 2000, 2000SP1, 2005, 2008, 2008R2, and 2012.
Protocols	MySQL protocol version 10. For information about the MySQL protocol, see MySQL Client/Server Protocol	Tabular Data Stream (TDS) protocol version 7.1 and higher. For information about the TDS protocol, see Tabular Data Stream Protocol
Authentication Methods	MySQL native authentication is supported.	SQL server authentication and Windows Authentication (Kerberos/NTLM) are supported.

Character Sets

The DataStream feature supports only the UTF-8 charset.

The character set used by the client while sending a request may be different from the character set used in the database server responses. Although the charset parameter is set during the connection establishment, it can be changed at any time by sending an SQL query. The character set is associated with a connection, and therefore, requests on connections with one character set cannot be multiplexed onto a connection with a different character set.

The NetScaler appliance parses the queries sent by the client and the responses sent by the database server.

The character set associated with a connection can be changed after the initial handshake by using the following two queries:

```
SET NAMES <charset> COLLATION <collation>

SET CHARACTER SET <charset>
<!--NeedCopy-->
```

Transactions

In MySQL, transactions are identified by using the connection parameter AUTOCOMMIT or the BEGIN:COMMIT queries. The AUTOCOMMIT parameter can be set during the initial handshake, or after the connection is established by using the query SET AUTOCOMMIT.

The NetScaler appliance explicitly parses each and every query to determine the beginning and end of a transaction.

In MySQL protocol, the response contains two flags to indicate whether the connection is a transaction, the TRANSACTION and AUTOCOMMIT flags.

If the connection is a transaction, the TRANSACTION flag is set. Or, if the AutoCommit mode is OFF, the AUTOCOMMIT flag is not set. The ADC appliance parses the response, and if either the TRANSACTION flag is set or the AUTOCOMMIT flag is not set, it does not do connection multiplexing. When these conditions are no longer true, the ADC appliance begins connection multiplexing.

> **Note**
>
> Transactions are also supported for MS SQL.

Special Queries

There are special queries, such as SET and PREPARE, that modify the state of the connection and may break request switching, and therefore, these need to be handled differently.

On receiving a request with special queries, the NetScaler appliance sends an OK response to the client and additionally, stores the request in the connection.

When a non-special query, such as INSERT and SELECT, is received along with a stored query, the NetScaler appliance first, looks for the server-side connection on which the stored query has already been sent to the database server. If no such connections exist, the ADC appliance creates a new connection, and sends the stored query first, and then, sends the request with the non-special query.

In case of SET, USE db, and INIT_DB special queries, the appliance modifies a field in the server side connection corresponding to the special query. This results in better reuse of the server side connection.

Only 16 queries are stored in each connection.

The following is a list of the special queries for which the ADC appliance has a modified behavior.

- SET query

  The SET SQL queries define variables that are associated with the connection. These queries are also used to define global variables, but as of now, the ADC appliance is unable to differentiate between local and global variables. For this query, the ADC appliance uses the 'store and forward' mechanism.

- USE <`db`> query

  Using this query, the user can change the database associated with a connection. In this case, the ADC appliance parses the <`db`> value sent and modifies a field in the server side connection to reflect the new database to be used.

- INIT_DB command

  Using this query, the user can change the database associated with a connection. In this case, ADC appliance parses the value sent and modifies a field in the server side connection to reflect the new database to be used.

- COM_PREPARE

  The ADC appliance stops request switching on receiving this command.

- PREPARE query

  This query is used to create prepared statements that are associated with a connection. For this query, the ADC appliance uses the 'store and forward' mechanism.

Audit Log Message Support

You can now configure the NetScaler appliance to generate audit log messages for the DataStream feature. Audit log messages are generated when client-side and server-side connections are established, closed, or dropped. The categories of messages that you can log and view are ERROR and INFO. Error messages for client-side connections begin with "CS" and error messages for server-side connections begin with "SS." Additional information is provided where necessary. For example, log messages for closed connections (CS_CONN_CLOSED) include only the connection ID. However, log messages for established connections (CS_CONN_ESTD) include information such as the user name, database name, and the client IP address in addition to the connection ID.

Domain Name System

September 21, 2020

You can configure the NetScaler appliance to function as an authoritative domain name server (ADNS server) for a domain. You can add the DNS resource records that belong to the domain for which the appliance is authoritative and configure resource record parameters. You can also configure the NetScaler appliance as a proxy DNS server that load balances a farm of DNS name servers that are either within your network or outside your network. You can configure the appliance as an end resolver and forwarder. You can configure DNS suffixes that enable name resolution when fully qualified domain names are not configured. The appliance also supports the DNS ANY query that retrieves all the records that belong to a domain.

You can configure the NetScaler appliance to concurrently function as an authoritative DNS server for one domain and a DNS proxy server for another domain. When you configure the NetScaler appliance as the authoritative DNS server or DNS proxy server for a zone, you can enable the appliance to use the Transmission Control Protocol (TCP) for response sizes that exceed the size limit specified for the User Datagram Protocol (UDP).

How DNS Works on the NetScaler

You can configure the NetScaler appliance to function as an ADNS server, DNS proxy server, end resolver, and forwarder. You can add DNS resource records on the NetScaler appliance, including service (SRV) records, IPv6 (AAAA) records, address (A) records, mail exchange (MX) records, canonical name (CNAME) records, pointer (PTR) records, start of authority (SOA) records, and text (TXT) records. Also, you can configure the NetScaler to load balance external DNS name servers.

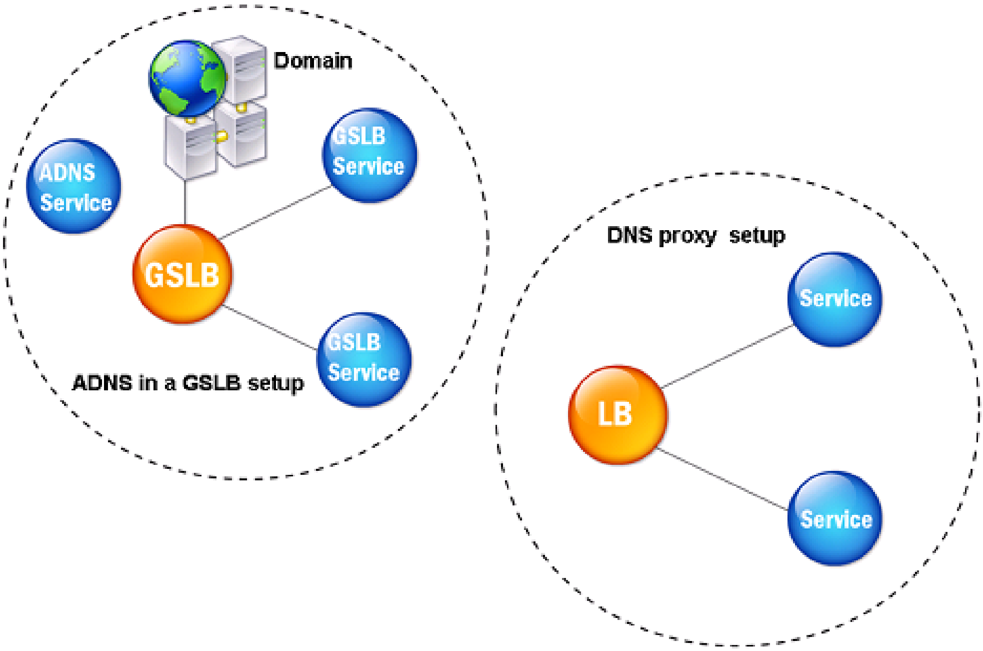
The NetScaler appliance can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

An ADNS server is a DNS server that contains complete information about a zone.

To configure the NetScaler appliance as an ADNS server for a zone, you must add an ADNS service, and then configure the zone. To do so, you add valid SOA and NS records for the domain. When a client sends a DNS request, the NetScaler appliance searches the configured resource records for the domain name. You can configure the ADNS service to be used with the NetScaler Global Server Load Balancing (GSLB) feature.

You can delegate a subdomain, by adding NS records for the subdomain to the zone of the parent domain. You can then make the NetScaler authoritative for the subdomain, by adding a “glue record” for each of the subdomain name servers. If GSLB is configured, theNetScaler makes a GSLB load balancing decision based on its configuration and replies with the IP address of the selected virtual server. The following figure shows the entities in an ADNS GSLB setup and a DNS proxy setup.

Figure 1. DNS Proxy Entity Model



The NetScaler appliance can function as a DNS proxy. Caching of DNS records, which is an important function of a DNS proxy, is enabled by default on the NetScaler appliance. This enables the NetScaler appliance to provide quick responses for repeated translations. You must also create a load balancing DNS virtual server, and DNS services, and then bind these services to the virtual server.

The NetScaler provides two options, minimum time to live (TTL) and maximum TTL for configuring the lifetime of the cached data. The cached data times out as specified by your settings for these two options. The NetScaler checks the TTL of the DNS record coming from the server. If the TTL is less than the configured minimum TTL, it is replaced with the configured minimum TTL. If the TTL is greater than the configured maximum TTL, it is replaced with the configured maximum TTL.

The NetScaler also allows caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called *negative caching*. Negative caching helps speed up responses to queries on a domain, and can optionally provide the record type.

A negative response can be one of the following:

- NXDOMAIN error message - If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler. The NetScaler caches the response locally, then returns the error message to the client.

- NODATA error message - The NetScaler sends a NODATA error message, if the domain name in query is valid but records of the given type are not available.

The NetScaler supports recursive resolution of DNS requests. In recursive resolution, the resolver (DNS client) sends a recursive query to a name server for a domain name. If the queried name server is authoritative for the domain, it responds with the requested domain name. Otherwise, the NetScaler queries the name servers recursively until the requested domain name is found.

Before you can apply the recursive query option, you must first enable it. You can also set the number of times the DNS resolver must send a resolution request (DNS retries) if a DNS lookup fails.

You can configure the NetScaler as a DNS forwarder. A forwarder passes DNS requests to external name servers. The NetScaler allows you to add external name servers and provides name resolution for domains outside the network. The NetScaler also allows you to set the name lookup priority to DNS or Windows Internet Name Service (WINS).

Round Robin DNS

When a client sends a DNS request to find the DNS resource record, it receives a list of IP addresses resolving to the name in the DNS request. The client then uses one of the IP addresses in the list, generally, the first record or IP address. Hence, a single server is used for the total TTL of the cache and is overloaded when a large number of requests arrive.

When the NetScaler receives a DNS request, it responds by changing the order of the list of DNS resource records in a round robin method. This feature is called *round robin DNS*. Round robin distributes the traffic equally between data centers. The NetScaler performs this function automatically. You do not have to configure this behavior.

Functional Overview

If the NetScaler is configured as an ADNS server, it returns the DNS records in the order in which the records are configured. If the NetScaler is configured as a DNS proxy, it returns the DNS records in the order in which it receives the records from the server. The order of the records present in the cache matches the order in which records are received from the server.

The NetScaler then changes the order in which records are sent in the DNS response in a round robin method. The first response contains the first record in sequence, the second response contains the second record in sequence, the third response contains the third record in sequence, and the order continues in the same sequence. Thus, clients requesting the same name can connect to different IP addresses.

Round Robin DNS Example

As an example of round robin DNS, consider DNS records that have been added as follows:

```
add dns addRec ns1 1.1.1.1  add dns addRec ns1 1.1.1.2  add dns addRec
    ns1 1.1.1.3  add dns addRec ns1 1.1.1.4
<!--NeedCopy-->
```

The domain, abc.com is linked to an NS record as follows:

```
add dns nsrec abc.com. ns1
<!--NeedCopy-->
```

When the NetScaler receives a query for the A record of ns1, the Address records are served in a round robin method as follows. In the first DNS response, 1.1.1.1 is served as the first record:

```
  ns1.                        1H IN A          1.1.1.1  ns1.
                            1H IN A          1.1.1.2  ns1.
                            1H IN A          1.1.1.3  ns1.
                            1H IN A          1.1.1.4
<!--NeedCopy-->
```

In the second DNS response, the second IP address, 1.1.1.2 is served as the first record:

```
  ns1.                        1H IN A          1.1.1.2  ns1.
                            1H IN A          1.1.1.3  ns1.
                            1H IN A          1.1.1.4  ns1.
                            1H IN A          1.1.1.1
<!--NeedCopy-->
```

In the third DNS response, the third IP address, 1.1.1.2 is served as the first record:

```
  ns1.                        1H IN A          1.1.1.3  ns1.
                            1H IN A          1.1.1.4  ns1.
                            1H IN A          1.1.1.1  ns1.
                            1H IN A          1.1.1.2
<!--NeedCopy-->
```

Configure DNS resource records

September 21, 2020

You configure resource records on the NetScaler ADC appliance when you configure the appliance as an ADNS server for a zone. You can also configure resource records on the appliance if the resource records belong to a zone for which the appliance is a DNS proxy server. On the appliance, you can configure the following record types:

- Service records
- AAAA records
- Address records
- Mail Exchange records
- Name Server records
- Canonical records
- Pointer records
- NAPTR records
- Start of Authority records
- Text records

The following table lists the record types that you can configure for a domain name record on the NetScaler appliance. For example, you can configure a maximum of 25 IP addresses for one record.

Table 1. Record Type and Number Configurable

Record Type	Number of Records
Address (A)	25
IPv6 (AAAA)	5
Mail exchange (MX)	12
Name server (NS)	16
Service (SRV)	8
Pointer (PTR)	20
Canonical name (CNAME)	1
Start of Authority (SOA)	1
Text (TXT)	20
Naming Authority Pointer (NAPTR)	20

> **Note:**
>
> The maximum number of IP addresses for a specific hostname is 25. However, the number of different address records can be more than 25.

Create SRV records for a service

September 21, 2020

The SRV record provides information about the services available on the NetScaler appliance. An SRV record contains the following information: name of the service and the protocol, domain name, TTL, DNS class, priority of the target, weight of records with the same priority, port of the service, and host name of the service. The NetScaler chooses the SRV record that has the lowest priority setting first. If a service has multiple SRV records with the same priority, clients use the weight field to determine which host to use.

Add an SRV record by using the CLI

At the command prompt, type the following commands to add an SRV record and verify the configuration:

```
-  add dns srvRec <domain> <target> -priority <positive_integer> -weight <positive_integer> -port <positive_integer> [-TTL <secs>]
-  sh dns srvRec <domain>
<!--NeedCopy-->
```

Example:

```
> add dns srvRec _http._tcp.example.com nameserver1.com -priority 1 -weight 1 -port 80
 Done
> show dns srvRec _http._tcp.example.com
1)      Domain Name : _http._tcp.example.com
        Target Host : nameserver1.com
        Priority : 1    Weight : 1
        Port : 80       TTL : 3600 secs
 Done
<!--NeedCopy-->
```

Modify or remove an SRV record by using the CLI

- To modify an SRV record, type the `set dns srvRec` command, the name of the domain for which the SRV record is configured, the name of the target host that hosts the associated service, and the parameters to be changed, with their new values.
- To remove an SRV record, type the `rm dns srvRec` command, the name of the domain for which the SRV record is configured, and the name of the target host that hosts the associated service.

Configure an SRV record by using the GUI

Navigate to Traffic Management > DNS > Records > SRV Records and create an SRV record.

Creating AAAA Records for a Domain Name

September 21, 2020

An AAAA resource record stores a single IPv6 address.

Add an AAAA record by using the CLI

At the command prompt, type the following commands to add an AAAA record and verify the configuration:

```
-  add dns aaaaRec <hostName> <IPv6Address> ... [-TTL <secs>]
-  show dns aaaaRec <hostName>
<!--NeedCopy-->
```

Example:

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57
   ab
 Done
> show dns aaaaRec www.example.com
1)      Host Name : www.example.com
        Record Type : ADNS              TTL : 5 secs
        IPV6 Address : 2001:db8::1428:57ab
 Done
<!--NeedCopy-->
```

To remove an AAAA record and all of the IPv6 addresses associated with the domain name, type the `rm dns aaaaRec` command and the domain name for which the AAAA record is configured. To remove only a subset of the IPv6 addresses associated with the domain name in an AAAA record, type the
`rm dns aaaaRec` command, the domain name for which the AAAA record is configured, and the IPv6 addresses that you want to remove.

Add an AAAA record by using the GUI

Navigate to Traffic Management > DNS > Records > AAAA Records and create an AAAA record.

Creating address records for a Domain Name

September 21, 2020

Address (A) records are DNS records that map a domain name to an IPv4 address.

You cannot delete Address records for a host participating in global server load balancing (GSLB). However, the NetScaler deletes Address records added for GSLB domains when you unbind the domain from a GSLB virtual server. Only user-configured records can be deleted manually. You cannot delete a record for a host referenced by records such as NS, MX, or CNAME.

Add an Address record by using the CLI

At the command prompt, type the following commands to add an Address record and verify the configuration:

```
-  add dns addRec <hostName> <IPAddress> [-TTL <secs>]
-  show dns addRec <hostName>
<!--NeedCopy-->
```

Example:

```
> add dns addRec ns.example.com 192.0.2.0
 Done
> show dns addRec ns.example.com
1)      Host Name : ns.example.com
        Record Type : ADNS                      TTL : 5 secs
        IP Address : 192.0.2.0
 Done
<!--NeedCopy-->
```

To remove an Address record and all of the IP addresses associated with the domain name, type the `rm dns addRec` command and the domain name for which the Address record is configured. To remove only a subset of the IP addresses associated with the domain name in an Address record, type the
`rm dns addRec` command, the domain name for which the Address record is configured, and the IP addresses that you want to remove.

Add an Address record by using the GUI

Navigate to Traffic Management > DNS > Records > Address Records and create an Address record.

Create MX records for a mail exchange server

September 21, 2020

Mail Exchange (MX) records are used to direct email messages across the Internet. An MX record contains an MX preference that specifies the MX server to be used. The MX preference values range from 0 through 65536. An MX record contains a unique MX preference number. You can set the MX preference and the TTL values for an MX record.

When an email message is sent through the Internet, a mail transfer agent sends a DNS query requesting the MX record for the domain name. This query returns a list of host names of mail exchange servers for the domain, along with a preference number. If there are no MX records, the request is made for the Address record of that domain. A single domain can have multiple mail exchange servers.

Add an MX record by using the CLI

At the command prompt, type the following commands to add an MX record and verify the configuration:

```
-  add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <
   secs>]
-  show dns mxRec <domain>
<!--NeedCopy-->
```

Example:

```
> add dns mxRec example.com -mx mail.example.com -pref 1
 Done
> show dns mxRec example.com
1)      Domain : example.com    MX Name : mail.example.com
        Preference :   1        TTL : 5 secs
 Done
<!--NeedCopy-->
```

Modify or remove an MX record by using the CLI

- To modify an MX record, type the set dns mxRec command, the name of the domain for which the MX record is configured, the name of the MX record, and the parameters to be changed, with their new values.
- To set the TTL parameter to its default value, type the unset dns mxRec command, the name of the domain for which the MX record is configured, the name of the MX record, and -TTL without any TTL value. You can use the unset dns mxRec command to unset only the TTL parameter.

- To remove an MX record, type the rm dns mxRec command, the name of the domain for which the MX record is configured, and the name of the MX record.

Add an MX record by using the GUI

Navigate to Traffic Management > DNS > Records > Mail Exchange Records and create an MX record.

Create NS Records for an authoritative server

September 21, 2020

Name Server (NS) records specify the authoritative server for a domain. You can configure a maximum of 16 NS records. You can use an NS record to delegate the control of a subdomain to a DNS server.

Create an NS record by using the CLI

At the command prompt, type the following commands to create an NS record and verify the configuration:

```
-  add dns nsRec <domain> <nameServer> [-TTL <secs>]
-  show dns nsRec <domain>
<!--NeedCopy-->
```

Example:

```
> add dns nsRec example.com nameserver1.example.com
 Done
> show dns nsRec example.com
1)      Domain : example.com    NameServer : nameserver1.example.com
        TTL : 5 sec
 Done
<!--NeedCopy-->
```

To remove an NS record, type the `rm dns nsRec` command, the name of the domain to which the NS record belongs, and the name of the name server.

Create an NS record by using the GUI

Navigate to Traffic Management > DNS > Records > Name Server Records and create an NS record.

Create CNAME records for a subdomain

September 21, 2020

A canonical name record (CNAME record) is an alias for a DNS name. These records are useful when multiple services query the DNS server. The host that has an address (A) record cannot have a CNAME record.

In some cases, a NetScaler appliance in proxy mode requests an address record from the cache instead of the server.

Add a CNAME record by using the CLI

At the command prompt, type the following commands to create a CNAME record and verify the configuration:

```
-  add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
-  show dns cnameRec <aliasName>
<!--NeedCopy-->
```

Example:

```
> add dns cnameRec www.example.com www.examplenw.com
 Done
> show dns cnameRec www.example.com
        Alias Name      Canonical Name  TTL
1)      www.example.com        www.examplenw.com       5 secs
 Done
<!--NeedCopy-->
```

To remove a CNAME record for a given domain, type the
`rm dns cnameRec` command and the alias of the domain name.

Add a CNAME record by using the GUI

Navigate to Traffic Management > DNS > Records > Canonical Records and create a CNAME record.

Cache CNAME records

NetScaler when deployed in a proxy mode does not always send the query for an address record to the back-end server. This happens when for a answer to a query for an address record, a partial CNAME chain is present in the cache. There are few conditions in which the ADC caches the partial CNAME record and serves the query from the cache. Following are the conditions:

- NetScaler should be deployed in a proxy mode
- The response from the back-end server should have a CNAME chain, for which the record type of last entry in the answer section must be a CNAME and the question type not a CNAME
- The response from the back-end server cannot be a No-data or NX-Domain
- The response from the back-end server has to be a authoritative response

Create NAPTR records for telecommunications domain

September 21, 2020

NAPTR (Naming Address Pointer) is one of the most commonly used DNS record in telecommunications domain. NAPTR records map the Internet telephony address space to the Internet address space. They therefore enable a mobile device to send a request to the correct server. The combination of NAPTR records with Service Records (SRV) allows the chaining of multiple records to form complex rewrite rules that produce new domain labels or uniform resource identifiers (URIs). The DNS code for NAPTR is 35.

NetScalers support NAPTR in two modes: ADNS mode and proxy mode. In proxy mode, the ADC caches the response from the servers and uses the cached records to server future queries. A maximum of 20 NAPTR records can be added for a particular domain in NetScaler. NetScaler caches the reply to a DNS NAPTR record query. Any subsequent requests for the NAPTR record is served from the cache.

Create a NAPTR record by using CLI

At the command prompt, type the following commands to add a NAPTR record and verify the configuration:

```
add dns naptrRec <order> <preference>[flags<string>][services<string>](
regexp<expressions>|-replacement<string>)[-TTL<secs>]
```

Remove a NAPTR record by using CLU

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services
 <string>] (-regexp <expression> | -replacement <string>))| -recordId <
positive_integer>@)
```

Configure a NAPTR record using GUI

Navigate to Traffic Management > DNS > Records > NAPTR Records and create an NAPTR record.

Create PTR records for IPv4 and IPv6 addresses

September 21, 2020

A pointer (PTR) record translates an IP address to its domain name. IPv4 PTR records are represented by the octets of an IP address in reverse order with the string “in-addr.arpa.” appended at the end. For example, the PTR record for the IP address 1.2.3.4 is 4.3.2.1.in-addr.arpa.

IPv6 addresses are reverse mapped under the domain IP6.ARPA. IPv6 reverse-maps use a sequence of nibbles separated by dots with the suffix “.IP6.ARPA” as defined in RFC 3596. For example, the reverse lookup domain name corresponding to the address, 4321:0:1:2:3:4:567:89ab would be b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

Add a PTR record by using the CLI

At the command prompt, type the following commands to add a PTR record and verify the configuration:

```
-  add dns ptrRec <reverseDomain> <domain> [-TTL <secs>]
-  show dns ptrRec <reverseDomain>
<!--NeedCopy-->
```

Example:

```
> add dns ptrRec 0.2.0.192.in-addr.arpa example.com
 Done
> show dns ptrRec 0.2.0.192.in-addr.arpa
1)      Reverse Domain Name : 0.2.0.192.in-addr.arpa
        Domain Name : example.com               TTL : 3600 secs
 Done
<!--NeedCopy-->
```

To remove a PTR record, type the `rm dns ptrRec` command and the reverse domain name associated with the PTR record

Add a PTR record by using the GUI

Navigate to Traffic Management > DNS > Records > PTR Records and create a PTR record.

Create SOA records for authoritative information

September 21, 2020

A Start of Authority (SOA) record is created only at the zone apex and contains information about the zone. The record includes, among other parameters, the primary name server, contact information (e-mail), and default (minimum) time-to-live (TTL) values for records.

Create an SOA record by using the CLI

At the command prompt, type the following commands to add an SOA record and verify the configuration:

```
-  add dns soaRec <domain\> -originServer \<originServerName\> -contact \<contactName\>
-  sh dns soaRec \<do main\>
<!--NeedCopy-->
```

Example:

```
> add dns soaRec example.com -originServer nameserver1.example.com -contact admin.example.com
 Done
> show dns soaRec example.com
1)      Domain Name : example.com
        Origin Server : nameserver1.example.com
        Contact : admin.example.com
        Serial No. : 100        Refresh : 3600 secs     Retry : 3 secs
        Expire : 3600 secs      Minimum : 5 secs        TTL : 3600 secs
 Done
<!--NeedCopy-->
```

Modify or remove an SOA record by using the CLI

- To modify an SOA record, type the `set dns soaRec` command, the name of the domain for which the record is configured, and the parameters to be changed, with their new values.
- To remove an SOA record, type the `rm dns soaRec` command and the name of the domain for which the record is configured.

Configure an SOA record by using the GUI

Navigate to Traffic Management > DNS > Records > SOA Records and create an SOA record.

Create TXT records for holding descriptive text

September 21, 2020

Domain hosts store TXT records for informative purposes. A TXT record's RDATA component, which consists of one or more character strings of variable length, can store practically any information that a recipient might need to know about the domain, including information about the service provider, contact person, email addresses, and associated details. SPF (Sender Policy Framework) protection has been the most prominent use case for the TXT record.

All configuration types (authoritative DNS, DNS proxy, end resolver, and forwarder configurations) on the NetScaler appliance support TXT records. You can add a maximum of 20 TXT resource records to a domain. Each resource record is stored with a unique, internally generated record ID. You can view the ID of a record and use it to delete the record. However, you cannot modify a TXT resource record.

Create a TXT resource record by using the CLI

At the command prompt, type the following commands to create a TXT resource record and verify the configuration:

```
-  add dns txtRec <domain> <string> ... [-TTL <secs>]
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

Example:

```
> add dns txtRec www.example.com "Contact: Mark" "Email: mark@example.
   com" -TTL 36000
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 13783       TTL : 36000 secs
    Record Type : ADNS
        "Contact: Mark"
        "Email: mark@example.com"
 Done
<!--NeedCopy-->
```

Remove a TXT resource record by using the CLI

At the command prompt, type the following commands to remove a TXT resource record and verify the configuration:

```
-  rm dns txtRec <domain> (<string> ... | -recordId <positive_integer>)
-  show dns txtRec [<domain> | -type <type>]
<!--NeedCopy-->
```

Example:
You can use the show dns txtRec command first to view the record ID of the TXT resource record that you want to remove, as shown:

```
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 36865       TTL : 36000 secs
    Record Type : ADNS
        "Contact: Evan"
        "Email: evan@example.com"
2)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
    Record Type : ADNS
        "Contact: Mark"
        "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

The simpler method of deleting a TXT record is to use the record ID. If you want to provide the strings, enter them in the order in which they are stored in the record. In the following example, the TXT record is deleted by using its record ID.

```
>rm dns txtRec www.example.com -recordID 36865
 Done
> show dns txtRec www.example.com
1)  Domain : www.example.com    Record id: 14373       TTL : 36000 secs
    Record Type : ADNS
        "Contact: Mark"
        "Email: mark1@example.com"
 Done
<!--NeedCopy-->
```

Configure a TXT record by using the GUI

Navigate to Traffic Management > DNS > Records > TXT Records and create a TXT record.

Viewing DNS statistics

September 21, 2020

You can view the DNS statistics generated by the NetScaler appliance. The DNS statistics include runtime, configuration, and error statistics.

View DNS records statistics by using the CLI

At the command prompt, type:

`stat dns`

Example:

```
> stat dns
DNS Statistics

Runtime Statistics
Dns queries                                    21
NS queries                                      8
SOA queries                                    18
.
.
.
Configuration Statistics
AAAA records                                   17
A records                                      36
MX records                                      9
.
.
.
Error Statistics
Nonexistent domain                             17
No AAAA records                                 0
No A records                                   13
.
.
.
Done
<!--NeedCopy-->
```

View DNS records statistics by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Statistics.

Configure a DNS zone

September 21, 2020

A DNS zone entity on the NetScaler ADC appliance facilitates the ownership of a domain on the appliance. A zone on the appliance also enables you to implement DNS Security Extensions (DNSSEC) for the zone, or to offload the zone's DNSSEC operations from the DNS servers to the appliance. DNSSEC sign operations are performed on all the resource records in a DNS zone. Therefore, if you want to sign a zone, or if you want to offload DNSSEC operations for a zone, you must first create the zone on the NetScaler appliance.

You must create a DNS zone on the appliance in the following scenarios:

- The NetScaler appliance owns all the records in a zone, that is, the appliance is operating as the authoritative DNS server for the zone. The zone must be created with the proxyMode parameter set to NO.
- The NetScaler appliance owns only a subset of the records in a zone, and all the other resource records in the zone are hosted on a set of back-end name servers for which the appliance is configured as a DNS proxy server. A typical configuration where the NetScaler appliance owns only a subset of the resource records in the zone is a global server load balancing (GSLB) configuration. Only the GSLB domain names are owned by the NetScaler appliance, while all the other records are owned by the back-end name servers. The zone must be created with the proxyMode parameter set to YES.
- You want to offload DNSSEC operations for a zone from your authoritative DNS servers to the appliance. The zone must be created with the proxyMode parameter set to YES. You might need to configure additional settings for the zone.

The current topic describes how to create a zone for the first two scenarios. For more information about how to configure a zone for offloading DNSSEC operations to the appliance, see Offload DNSSEC operations to the NetScaler appliance.

> **Note**
>
> If the NetScaler is operating as the authoritative DNS server for a zone, you must create Start of Authority (SOA) and name server (NS) records for the zone before you create the zone. If the NetScaler is operating as the DNS proxy server for a zone, SOA and NS records must not be created on the NetScaler appliance. For more information about creating SOA and NS records, see Configure DNS resource records.

When you create a zone, all existing domain names and resource records that end with the name of the zone are automatically treated as a part of the zone. Additionally, any new resource records created with a suffix that matches the name of the zone are implicitly included in the zone.

Create a DNS zone on the NetScaler appliance by using the CLI

At the command prompt, type the following command to add a DNS zone to the NetScaler appliance and verify the configuration:

```
-  add dns zone <zoneName> -proxyMode ( YES | NO )
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

Example:

```
> add dns zone example.com -proxyMode Yes
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : YES
Done
<!--NeedCopy-->
```

Modify or remove a DNS zone by using the CLI

- To modify a DNS zone, type the `set dns zone` command, the name of the DNS zone, and the parameters to be changed, with their new values.
- To remove a DNS zone, type the `rm dns zone` command and the name of the dns zone.

Configure a DNS zone by using the GUI

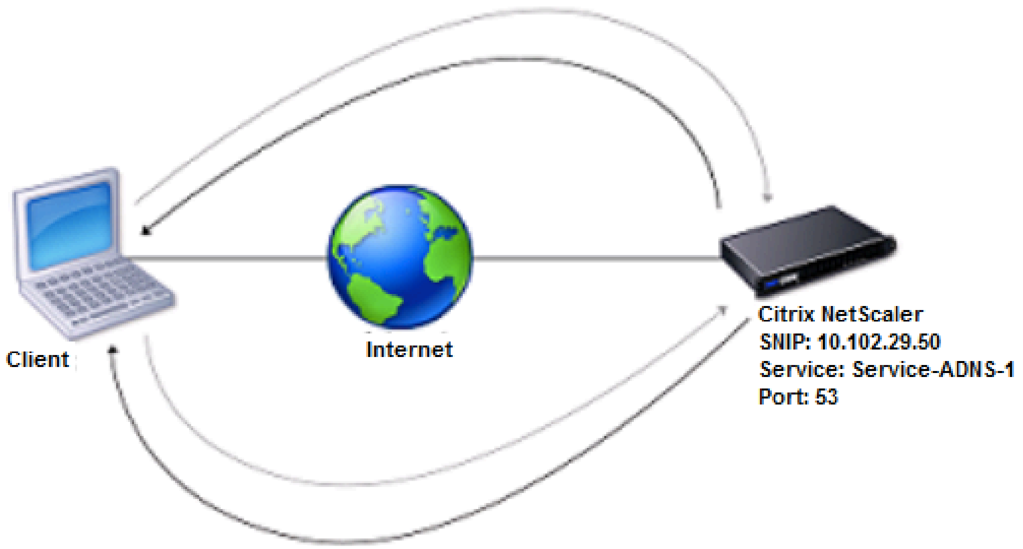
Navigate to Traffic Management > DNS > Zones and create a DNS zone.

Configure the NetScaler as an authoritative domain name server

September 21, 2020

You can configure the NetScaler appliance to function as an authoritative domain name server (ADNS) for a domain. As an ADNS server for a domain, the NetScaler resolves DNS requests for all types of DNS records that belong to the domain. To configure the NetScaler to function as an ADNS server for a domain, you must create an ADNS service and configure NS and Address records for the domain on the NetScaler. The ADNS service can be configured using the subnet IP address (SNIP) or a separate IP address. The following topology diagram shows a sample configuration and the flow of requests and responses.

Figure 1. NetScaler as an ADNS



The following table shows the parameters that are configured for the ADNS service illustrated in the preceding topology diagram.

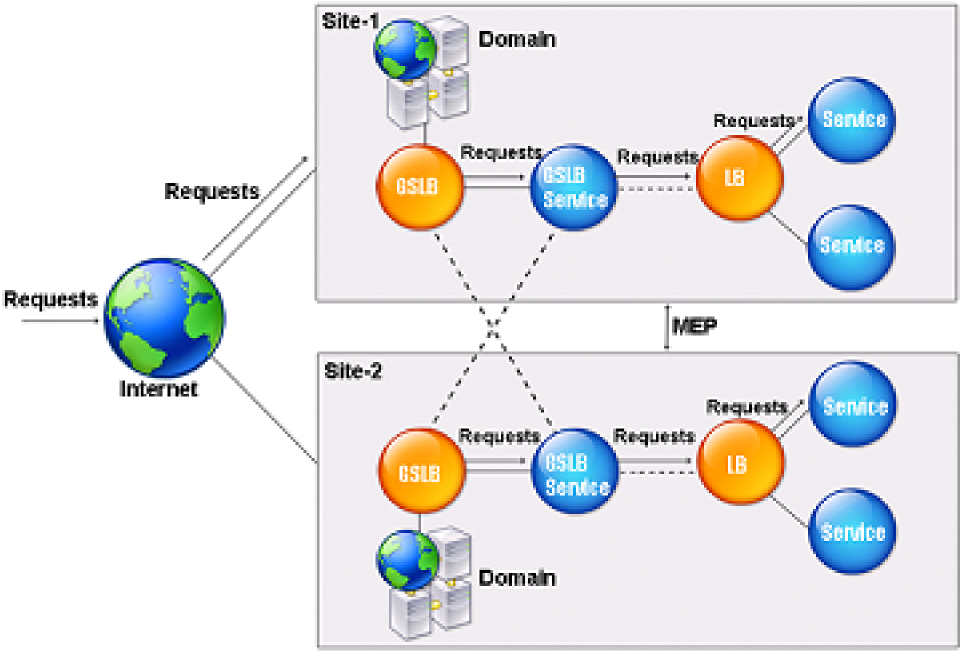
Entity type	Name	IP address	Type	Port
ADNS Service	Service-ADNS-1	10.102.29.51	ADNS	53

Table 1. Example of ADNS Service Configuration

To configure an ADNS setup, you must configure the ADNS service. For instructions on configuring the ADNS service, see "Load balancing".

During DNS resolution, the ADNS server directs the DNS proxy or local DNS server to query the NetScaler for the IP address of the domain. Because the NetScaler is authoritative for the domain, it sends the IP address to the DNS proxy or local DNS server. The following diagram describes the placement and role of the ADNS server in a GSLB configuration.

Figure 2. GSLB Entity Model


Site-1
Domain
Requests
GSLB
GSLB Service
LB
Service
Service
Requests
Requests
Internet
MEP
Site-2
GSLB
GSLB Service
LB
Service
Service
Domain

Note: In ADNS mode, if you remove SOA and ADNS records, the following do not function for the domain hosted by the NetScaler: ANY query (for more information about the ANY query, see DNS ANY query, and negative responses, such as NODATA and NXDOMAIN.

Create an ADNS service

An ADNS service is used for global service load balancing. For more information about creating a GSLB setup, see "Global server load balancing". You can add, modify, enable, disable, and remove an ADNS service. For instructions on creating an ADNS service, see Configure services.

Note: You can configure the ADNS service to use SNIP or any new IP address.

When you create an ADNS service, the NetScaler responds to DNS queries on the configured ADNS service IP and port.

You can verify the configuration by viewing the properties of the ADNS service. You can view properties such as name, state, IP address, port, protocol, and maximum client connections.

Configure the ADNS setup to use TCP

By default, some clients use the User Datagram Protocol (UDP) for DNS, which specifies a limit of 512 bytes for the payload length of UDP packets. To handle payloads that exceed 512 bytes in size, the client must use the TCP. To enable DNS communications over TCP, you must configure the NetScaler appliance to use the TCP protocol for DNS. The NetScaler then sets the truncation bit in the DNS response packets. The truncation bit specifies that the response is too large for UDP and that the client must send the request over a TCP connection. The client then uses the TCP protocol on port 53 and opens a new connection to the NetScaler. The NetScaler listens on port 53 with the IP address of the ADNS service to accept the new TCP connections from the client.

To configure the NetScaler to use the TCP protocol, you must configure an ADNS_TCP service. For instructions on creating an ADNS_TCP service, see
Load balancing.

> **Important**
>
> To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure the ADNS and ADNS_TCP services. The IP address of the ADNS_TCP service must be same as the IP address of the ADNS service.

Add DNS resource records

After you create an ADNS service, you can add DNS records. For instructions on adding DNS records, see Configure DNS resource records.

Remove ADNS services

For instructions on removing services, see Load balancing.

Configure domain delegation

Domain delegation is the process of assigning responsibility for a part of the domain space to another name server. Therefore, during domain delegation, the responsibility for responding to the query is delegated to another DNS server. Delegation uses NS records.

In the following example, sub1.abc.com is the subdomain for abc.com. The procedure describes the steps to delegate the subdomain to the name server ns2.sub1.abc.com and add an Address record for ns2.sub1.abc.com.

To configure domain delegation, you need to perform the following tasks, which are described in the sections that follow:

1. Create an SOA record for a domain.

2. Create an NS record to add a name server for the domain.
3. Create an Address record for the name server.
4. Create an NS record to delegate the subdomain.
5. Create a glue record for the name server.

Create an SOA record

For instructions on configuring SOA records, see Create SOA records for authoritative information.

Create an NS record for a name server

For instructions on configuring an NS record, see Create NS records for an authoritative server. In the **Name Server** list, select the primary authoritative name server, for example, ns1.abc.com.

Create an address record

For instructions on configuring Address records, see Create address records for a domain name. In the Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns1.abc.com and 10.102.11.135, respectively.

Create an NS record for domain delegation

For instructions on configuring NS records, see Create NS records for an authoritative server. In the Name Server drop-down list, select the primary authoritative name server, for example, ns2.sub1.abc.com.

Create a glue record

NS records are usually defined immediately after the SOA record (but this is not a restriction.) A domain must have at least two NS records. If an NS record is defined within a domain, it must have a matching Address record. This Address record is referred to as a glue record. Glue records speed up DNS queries.

For instructions on adding glue records for a subdomain, see the procedure for adding an Address (A) record, Configure DNS resource records.

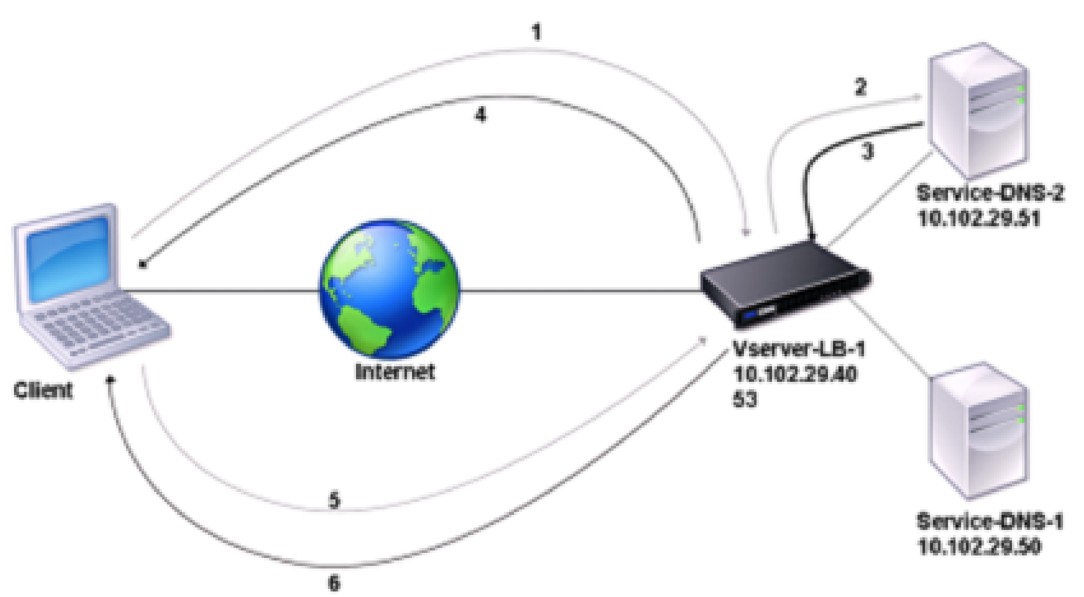
For instructions on configuring Address records, see [Create address records for a domain name]/(/en-us/netscaler/11-1/dns/configure-dns-resource-records/create-address-records.html). In Host Name and IP address text boxes, type the domain name for the DNS Address record and the IP address, for example, ns2.sub1.abc.com and 10.102.12.135, respectively.

Configure the NetScaler appliance as a DNS proxy server

September 21, 2020

As a DNS proxy server, the NetScaler ADC appliance can function as a proxy for either a single DNS server or a group of DNS servers. The flow of requests and responses is illustrated in the following sample topology diagram.

Figure 1. NetScaler as DNS proxy



By default, the NetScaler appliance caches responses from DNS name servers. When the appliance receives a DNS query, it checks for the queried domain in its cache. If the address for the queried domain is present in its cache, the NetScaler returns the corresponding address to the client. Otherwise, it forwards the query to a DNS name server that checks for the availability of the address and returns it to the NetScaler. The NetScaler then returns the address to the client.

For requests for a domain that has been cached earlier, the NetScaler serves the Address record of the domain from the cache without querying the configured DNS server.

The appliance discards a record stored in its cache when the time-to-live (TTL) value of the record reaches the configured value. A client that requests an expired record has to wait until the NetScaler retrieves the record from the server and updates its cache. To avoid this delay, the NetScaler proactively updates the cache by retrieving the record from the server before the record expires.

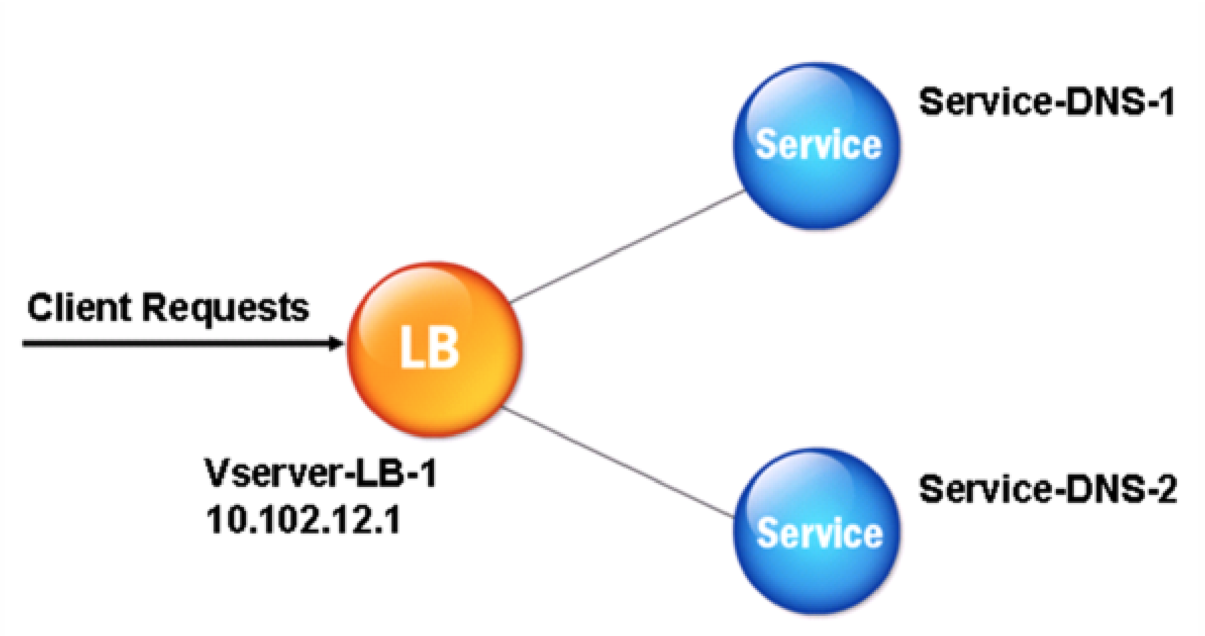
The following table lists sample names and the values of the entities that need to be configured on the NetScaler.

Table 1. Example of DNS Proxy Entity Configuration

Entity type	Name	IP address	Type	Port
LB virtual server	Vserver-DNS-1	10.102.29.40	DNS	53
Services	Service-DNS-1	10.102.29.50	DNS	53
Services	Service-DNS-2	10.102.29.51	DNS	53

The following diagram shows the entities of a DNS Proxy and the values of the parameters to be configured on the NetScaler.

Figure 2. DNS Proxy Entity Model



> **Note**
>
> To configure DNS proxy, you need to know how to configure load balancing services and virtual servers.

Create a load balancing virtual server

To configure a DNS Proxy on the NetScaler, configure a load balancing virtual server of type DNS. To configure a DNS virtual server to load balance a set of DNS servers that support recursive queries, you must set the Recursion Available option. With this option, the RA bit is set to ON in the DNS replies from the DNS virtual server.

For instructions on creating a load balancing virtual server, see "Load Balancing".

Create DNS services

After creating a load balancing virtual server of type DNS, you must create DNS services. You can add, modify, enable, disable, and remove a DNS service. For instructions on creating a DNS service, see "Load Balancing".

Bind a load balancing virtual server to DNS services

To complete the DNS Proxy configuration, you must bind the DNS services to the load balancing virtual server. For instructions on binding a service to a load balancing virtual server, see "Load Balancing".

Configure the DNS proxy setup to use TCP

Some clients use the User Datagram Protocol (UDP) for DNS communications. However, UDP specifies a maximum packet size of 512 bytes. When payload lengths exceed 512 bytes, the client must use the Transmission Control Protocol (TCP). When a client sends the NetScaler ADC appliance a DNS query, the appliance forwards the query to one of the name servers. If the response is too large for a UDP packet, the name server sets the truncation bit in its response to the NetScaler. The truncation bit indicates that the response is too large for UDP and that the client must send the query over a TCP connection. The NetScaler relays the response to the client with the truncation bit intact and waits for the client to initiate a TCP connection with the IP address of the DNS load balancing virtual server, on port 53. The client sends the request over a TCP connection. The NetScaler appliance then forwards the request to the name server and relays the response to the client.

To configure the NetScaler to use the TCP protocol for DNS, you must configure a load balancing virtual server and services, both of type DNS_TCP. You can configure monitors of type DNS_TCP to check the state of the services. For instructions on creating DNS_TCP virtual servers, services, and monitors, see "Load Balancing."

For updating the records proactively, the NetScaler uses a TCP connection to the server to retrieve the records.

> **Important** To configure the NetScaler to use UDP for DNS and use TCP only when the payload length of UDP exceeds 512 bytes, you need to configure DNS and DNS_TCP services. The IP address of the DNS_TCP service must be same as that of the DNS service.

Configure time-to-live values for DNS entries

The TTL is the same for all DNS records with the same domain name and record type. If the TTL value is changed for one of the records, the new value is reflected in all records of the same domain name and type. The default TTL value is 3600 seconds. The minimum is 0, and the maximum is 604800. If a DNS entry has a TTL value less than the minimum or greater than the maximum, it is saved as the minimum or maximum TTL value, respectively.

Specify the minimum and/or maximum TTL by using the CLI

At the NetScaler command prompt, type the following commands to specify the minimum and maximum TTL and verify the configuration:

```
-  set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -minTTL 1200 -maxTTL 1800
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        Minimum TTL: 1200              Maximum TTL: 1800
            .
            .
            .
 Done
>
<!--NeedCopy-->
```

Specify the minimum and/or maximum TTL by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in TTL, in the Minimum and Maximum text boxes, type the minimum and maximum time to live (in seconds), respectively, and then click OK.

Note: When the TTL expires, the record is deleted from the cache. The NetScaler proactively contacts the servers and obtains the DNS record just before the DNS record expires.

Flush DNS records

You can delete all DNS records present in the cache. For example, you might want to flush DNS records when a server is restarted after modifications are made.

Delete all proxy records by using the CLI

At the NetScaler command prompt, type:

`flush dns proxyRecords`

Delete all proxy records by using the GUI

1. Navigate to Traffic Management > DNS > Records.
2. In the details pane, click Flush Proxy Records.

Add DNS resource records

You can add DNS records to a domain for which the NetScaler ADC appliance is configured as a DNS proxy server. For information about adding DNS records, see Configuring DNS Resource Records.

Remove a load balancing DNS virtual server

For information about removing a load balancing virtual server, see Load Balancing.

Limit the number of concurrent DNS requests on a client connection

You can limit the number of concurrent DNS requests on a single client connection, which is identified by the <clientip:port>-<vserver ip:port> tuple. Concurrent DNS requests are those requests that the NetScaler appliance has forwarded to the name servers and for which the appliance is awaiting responses. Limiting the number of concurrent requests on a client connection enables you to protect the name servers when a hostile client attempts a Distributed Denial of Service (DDoS) attack by sending a flood of DNS requests. When the limit for a client connection is reached, subsequent DNS requests on the connection are dropped till the outstanding request count goes below the limit. This limit does not apply to the requests that the NetScNetScaleraler appliance serves out of its cache.

The default value for this parameter is 255. This default value is sufficient in most scenarios. If the name servers serve a large number of concurrent DNS requests under normal operating conditions, you can specify either a large value or a value of zero (0). A value of 0 disables this feature and specifies that there is no limit to the number of DNS requests that are allowed on a single client connection. This is a global parameter and applies to all the DNS virtual servers that are configured on the NetScaler appliance.

Specify the maximum number of concurrent DNS requests allowed on a single client connection by using the CLI

At the command prompt, type the following commands to specify the maximum number of concurrent DNS requests allowed on a single client connection and verify the configuration:

```
-  set dns parameter -maxPipeline <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -maxPipeline 1000
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
        .
        .
        .
        Max DNS Pipeline Requests: 1000
 Done
<!--NeedCopy-->
```

Specify the maximum number of concurrent DNS requests allowed on a single client connection by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, specify a value for Max DNS Pipeline Requests.
4. Click OK.

Configure the NetScaler as an end resolver

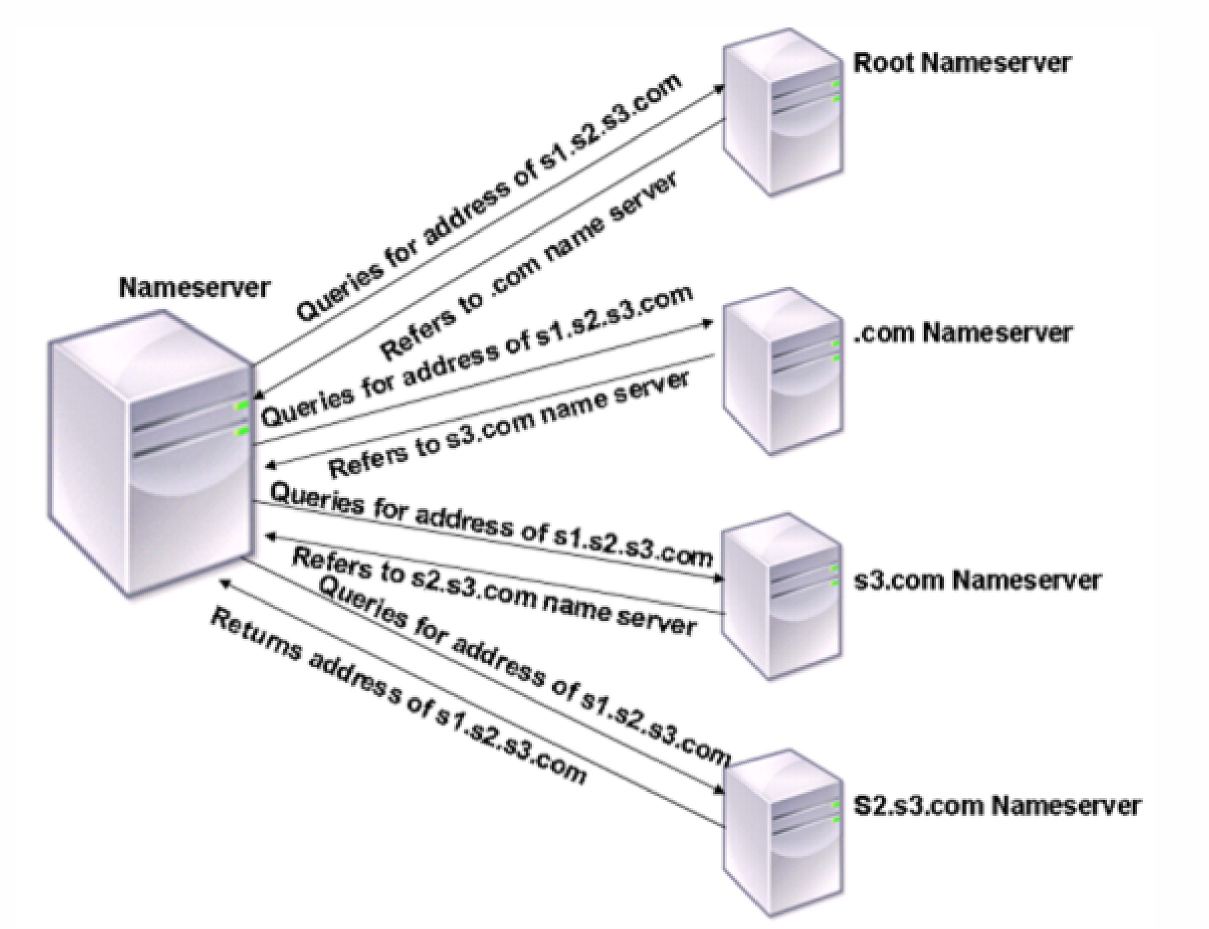
September 21, 2020

A resolver is a procedure that is invoked by an application program that translates a domain/host name to its resource record. The resolver interacts with the LDNS, which looks up the domain name to obtain its IP address. The NetScaler can provide end-to-end resolution for DNS queries.

In recursive resolution, the NetScaler appliance queries different name servers recursively to access the IP address of a domain. When the NetScaler receives a DNS request, it checks its cache for the DNS record. If the record is not present in the cache, it queries the root servers configured in the ns.conf file. The root name server reports back with the address of a DNS server that has detailed information about the second-level domain. The process is repeated until the required record is found.

When you start the NetScaler appliance for the first time, 13 root name servers are added to the ns.conf file. The NS and Address records for the 13 root servers are also added. You can modify the ns.conf file, but the NetScaler does not allow you to delete all 13 records; at least one name server entry is required for the appliance to perform name resolution. The following diagram illustrates the process of name resolution.

Figure 1. Recursive resolution


Root Nameserver
Nameserver
Queries for address of s1.s2.s3.com
Refers to .com name server
.com Nameserver
Queries for address of s1.s2.s3.com
Refers to s3.com name server
Queries for address of s1.s2.s3.com
s3.com Nameserver
Refers to s2.s3.com name server
Queries for address of s1.s2.s3.com
Returns address of s1.s2.s3.com
S2.s3.com Nameserver

In the process shown in the diagram, when the name server receives a query for the address of s1.s2.s3.com, it first checks the root name servers for s1.s2.s3.com. A root name server reports back with the address of the .com name server. If the address of s1.s2.s3.com is found in the name server, it responds with a suitable IP address. Otherwise, it queries other name servers for s3.com, then for s2.s3.com to retrieve the address of s1.s2.s3.com. In this way, resolution always starts from root name servers and ends with the domain's authoritative name server.

Note: For recursive resolution functionality, caching should be enabled.

Enable recursive resolution

To configure the NetScaler appliance to function as an end resolver, you must enable recursive resolution on the appliance.

Enable recursive resolution by using the CLI

At the command prompt, type the following commands to enable recursive resolution and verify the configuration:

```
-  set dns parameter -recursion ENABLED
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -recursion ENABLED
 Done
> show dns parameter
        DNS parameters:
            .
            .
            .
        Recursive Resolution : ENABLED
            .
            .
            .
 Done
<!--NeedCopy-->
```

Enable recursive resolution by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, select the Enable recursion check box, and then click OK.

Set the Number of Retries

The NetScaler appliance can be configured to make a preconfigured number of attempts (called DNS retries) when it does not receive a response from the server to which it sends a query. By default, the number of DNS retries is set to 5.

Set the number of DNS retries by using the CLI

At the command prompt, type the following commands to set the number of retries and verify the configuration:

```
-  set dns parameter -retries <positive_integer>
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set DNS parameter -retries 3
 Done
> show dns parameter
       DNS parameters:
       DNS retries: 3
           .
           .
           .
 Done
<!--NeedCopy-->
```

Set the number of retries by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, under Settings, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, in the DNS Retries text box, type the DNS resolver request retry count, and then click OK.

Configure the NetScaler appliance as a forwarder

September 21, 2020

A forwarder is a server that forwards DNS queries to DNS servers that are outside the forwarder server's network. Queries that cannot be resolved locally are forwarded to other DNS servers. A forwarder accumulates external DNS information in its cache as it resolves DNS queries. To configure the NetScaler appliance as a forwarder, you must add an external name server.

The NetScaler appliance allows you to add external name servers to which it can forward the name resolution queries that cannot be resolved locally. To configure the NetScaler appliance as a forwarder, you must add the name servers to which it must forward name resolution queries. You can specify the lookup priority to specify the name service that the NetScaler appliance must use for name resolution.

Add a name server

September 21, 2020

You can create a name server by specifying its IP address or by configuring an existing virtual server as the name server.

- **IP address-based name server** - An external name server to contact for domain name resolution. If multiple IP address-based name servers are configured on the appliance, and the local parameter is not set on any of them, incoming DNS queries are load balanced across all the name servers, in round robin fashion.
- **Virtual server-based name server** - A DNS virtual server configured in the NetScaler. If you want more fine-grained control on how external DNS name servers are load balanced (for example, you want a load balancing method other than round robin), do the following:
- Configure a DNS virtual server on the appliance
- Bind the external name servers as its services
- Specify the name of the virtual server in this command.

To verify the configuration, you can use the `show dns nameServer` command.

To remove a name server, at the NetScaler CLI, type the `rm dns nameServer` command followed by the IP address of the name server.

To view the details of the DNS nameserver, at the NetScaler CLI, type `show dns nameServer` command followed by the IP address of the name server.

Add a name server (when the NetScaler appliance acts as a forwarder) by using the CLI

At the command prompt, type;

```
add dns nameServer ((<IP>) | <dnsVserverName>)
<!--NeedCopy-->
```

or

```
add dns nameServer ((<IP> | <dnsVserverName>) [-type <type>]
<!--NeedCopy-->
```

Examples:

```
add dns nameServer dnsVirtualNS

add dns nameServer 192.0.2.11 -type TCP

add dns nameServer 192.0.2.12 -type UDP_TCP


add dns nameServer 192.0.2.10
show dns nameServer 192.0.2.10

1) 192.0.2.10  -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

> **Note:**
>
> If the name server type is not specified, a UDP name server is created by default. To create a name server of type TCP or UDP_TCP, you must specify the type.
>
> When you specify the type as UDP_TCP, two name servers (one UDP name server and one TCP name server) are created for the given IP address.

Add a name server (when the NetScaler appliance acts as a resolver) by using the CLI

At the command prompt, type:

```
add dns nameServer ((<IP> [-local]) | <dnsVserverName>)
<!--NeedCopy-->
```

Example:

```
add dns nameServer 10.102.9.19 -local
show dns nameServer
1)  10.102.9.19 LOCAL -  State: UP  Protocol: UDP
Done
<!--NeedCopy-->
```

local - Mark the IP address as one that belongs to a local recursive DNS server on the NetScaler appliance. The appliance recursively resolves queries received on an IP address that is marked as being local.
For recursive resolution to work, the global DNS parameter, `recursion`, must also be set.
If no name server is marked as being local, the appliance functions as a stub resolver and load balances the name servers.

Add a name server by using the GUI

Navigate to **Traffic Management** > **DNS** > **Name Servers** and create a name server.

Set DNS lookup priority

September 21, 2020

You can set the lookup priority to either DNS or WINS. This option is used in the SSL VPN mode of operation.

Set the lookup priority to DNS by using the CLI

At the command prompt, type the following commands to set the lookup priority to DNS and verify the configuration:

```
-  set dns parameter -nameLookupPriority (DNS | WINS)
-  show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -nameLookupPriority DNS
 Done
> show dns parameter
            .
            .
            .
        Name lookup priority : DNS
            .
            .
            .
 Done
<!--NeedCopy-->
```

Set lookup priority to DNS by using the GUI

1. Navigate to **Traffic Management > DNS**.
2. In the details pane, under **Settings**, click Change DNS settings.
3. In the **Configure DNS Parameters** dialog box, under **Name Lookup Priority**, select DNS or WINS, and then click OK.

> **Note**
>
> If the DNS virtual server that you have configured is DOWN and if you set the -nameLookupPriority to DNS, the NetScaler does not attempt WINS lookup. Therefore, if a DNS

> virtual server is not configured or is disabled, set the
> -nameLookupPriority to WINS.

Disable and enable name servers

September 21, 2020

The following procedure describes the steps to enable or disable an existing name server.

Enable or disable a name server by using the CLI

At the command prompt, type the following commands to enable or disable a name server and verify the configuration:

```
-  (enable | disable) dns nameServer <IPAddress>
-  show dns nameServer <IPAddress>
<!--NeedCopy-->
```

Example:

```
> disable dns nameServer 10.102.9.19
 Done
> show dns nameServer 10.102.9.19
1)       10.102.9.19: LOCAL  -  State: OUT OF SERVICE
 Done
<!--NeedCopy-->
```

Enable or disable a name server by using the GUI

1. Navigate to Traffic Management > DNS > Name Servers.
2. In the details pane, select the name server that you want to enable or disable.
3. Click Enable or Disable. If a name server is enabled, the Disable option is available. If a name server is disabled, the Enable option is available.

Configure DNS logging

September 21, 2020

You can configure the NetScaler appliance to log the DNS requests and responses that it handles. The appliance logs the DNS requests and responses in SYSLOG format. You can choose to log either DNS requests or DNS responses, or both, and send the syslog messages to a remote log server. The log messages can be used to:

- Audit the DNS responses to the client
- Audit DNS clients
- Detect and prevent DNS attacks
- Troubleshoot

A NetScaler appliance can log the following sections in the DNS request or response, on the basis of your configuration:

- Header Section
- Questions Section
- Answer Section
- Authority Section
- Additional Section

DNS profiles

You can use a DNS profile to configure various DNS parameters that you want the DNS endpoint to apply to the DNS traffic. In the profile, you can enable logging, caching and negative caching.

Important: From NetScaler 11.0 release, enabling DNS caching using global DNS parameters has been deprecated. You can enable or disable DNS caching using DNS profiles. You can now enable DNS caching for an individual virtual server by enabling DNS caching in a DNS profile and setting the DNS profile to the individual virtual server.

DNS profiles support the following types of DNS logging:

- DNS Query Logging
- DNS Answer Section Logging
- DNS Extended Logging
- DNS Error Logging

DNS query logging

You can configure a NetScaler appliance to log only the DNS queries that are received by the DNS endpoints on the appliance.

Note: If errors occur during processing of a query, they are logged if this option is set in the DNS profile.

Following is an example of a query log message:

```
DNS DNS_QUERY 143 0 :  U:10.102.27.70#61297:10.102.27.73#53/22142/Q/
(RD)/NO/1/0/0/0#test.com./1#
<!--NeedCopy-->
```

DNS answer section logging

You can configure a NetScaler appliance to log all the Answer sections in the DNS responses that appliance sends to the client. DNS Answer Section logging is very useful when NetScaler is configured as a DNS resolver, or in GLSB use cases.

Following is an example of a DNS answer section log:

```
DNS DNS_RESPONSE 6678 0 :  U:100.100.100.210#32776:100.100.100.10#
53/61373/Q/(RD,AA,RA,R)/NO/1/1/2/4#n1.citrix.com1./
28#ANS#AAAA/120/1111:2345:6789:ffab:abcd:effa:1234:3212##
<!--NeedCopy-->
```

DNS extended logging

To configure a NetScaler appliance to log Authority and Additional sections in the DNS responses, enable Extended logging with Answer Section logging.

Note: If errors occur during processing of either queries or responses, the errors are logged if this option is set in the DNS profile.

Following is an example of a message logged when the cache lookup is completed and the response is embedded in the packet:

```
DNS DNS_RESPONSE 2252 0 :  T:100.100.100.118#21411:100.100.100.10
#53/48537/Q/(RD,AA,CD,RA,R)/NO/1/1/2/6#a1.citrix.com1./1#ANS#A/
120/1.1.1.1##AUTH#citrix.com1/NS/120/n2.citrix.com1#n1.citrix.com1##ADD
    #n1.citrix.com1
/A/120/1.1.1.1#1.1.1.2##n1.citrix.com1/AAAA/120/
1111:2345:6789:ffab:abcd:effa:1234:3212##n2.citrix.com1/A/120/2.1.1.2
##n2.citrix.com1/AAAA/120/2222:faff:3212:8976:123:1241:64:ff9b##OPT
    /0/1280/DO##
<!--NeedCopy-->
```

DNS error logging

You can configure a NetScaler appliance to log the errors or failures that occur when it processes a DNS query or response. For these errors, the appliance logs the DNS header, Question sections and OPT records.

Following is an example of a message logged when an error occurs during processing of a DNS request or response:

```
DNS DNS_ERROR 149 0 :  U:10.102.27.70#27832:10.102.27.73#53/61153/Q/
(RD)/NO/1/0/0/0#test.com./1140#Packet Dropped
```

```
<!--NeedCopy-->
```

Policy based logging

You can configure custom logging based on DNS expressions by configuring the logAction on DNS policies, Rewrite, or Responder policies. You can specify that logging occurs only when a particular DNS policy evaluates to true. For more information, see Configure policy based logging for DNS.

Understand the NetScaler syslog log message format

NetScaler appliance log DNS requests and responses in the following Syslog format:

```
<transport> :<client IP>#<client ephemeral port>:<DNS endpoint IP>#<port>
 : <query id> /opcode/header flags/rcode/question section count/answer section count
/ auth section count / additional section count #<queried domain name>
/<queried type>#...
<!--NeedCopy-->
```

- **<transport>:**
    - T = TCP
    - U = UDP
- **<client IP>#< client ephemeral port >:**
  DNS client IP address and port number
- **<DNS endpoint IP>#<port>:**
  NetScaler DNS endpoint IP address and port number
- **<query id>:**
  Query ID
- **<opcode>:**
  Operation code. Supported Values:
    - Q : query
    - I : inverse query
    - S: status
    - X0: unassigned
    - N : notify
    - U : update
    - X1-10: unassigned values

- **<header flags>:**
  Flags. Supported Values:
  - RD : recursion desired
  - TC : truncated
  - AA : authoritative response
  - CD : check disabled
  - AD : authenticated data
  - Z : unassigned
  - RA : recursion available
  - R : response
- **<rcode>:**
  Response Code. Supported Values:
  - NO : no error
  - F format error
  - S : server failure
  - NX : non-existent domain
  - NI : not implemented
  - R: query refused
  - YX : Name Exists when it should not
  - YXR : RR Set Exists when it should not
  - NXR: RR Set that should exist does not
  - NAS : Server Not Authoritative for zone
  - NA : Not Authorized
  - NZ : Name not contained in zone
  - X1-5: unassigned
- **/question section count/answer section count/auth section count/additional section count:**
  Question section, Authority section count, and Additional section count in the DNS request
- **<queried domain name>/<queried type>:**
  Queried domain and queried type in the DNS request
- **#ANS#<record type>/<ttl>/.. #AUTH#<domain name>/<record type>/<ttl>.. #ADD#<domain name>/<record type>/<ttl>…:**
  In case of DNS responses:

  Answer Section is logged if answer section logging is enabled in the DNS profile.
  Authority and Additional sections are logged if extended logging is enabled in the DNS profile.
  The log format would differ depending on the type of record. For more information see Understanding the Record Logging Format.

  - ANS: answer section
  - AUTH: authority
  - ADD: Additional section

- **OPT/<edns version>/UDP max payload size/DO:**
  OPT record format in the DNS log
- **OPT/<EDNS version>/<UDP payload size>/<"DO"or empty based on whether DNSSEC OK bit is set or not>/<value of RDLEN>/ECS/<Q/R>/<option length>/<Family>/<Source Prefix-Length>/<Scope Prefix-Length>/<ECS Address>:**
  If the DNS query or response includes the EDNS Client Subnet (ECS) option, then that is also logged in the OPT record format in the DNS log file.

  When a DNS query with ECS option that includes either IPv4 or IPv6 address is sent, the ECS option is logged with either "ECS/Q" indicating that the values in the log are from the query or "ECS/R" indicating that the values in the log are from the response.

  The value of Scope Prefix-Length is also set appropriately. In case of the DNS Query, it is set to zero,and for response, it is set to the calculated value.

  The following table describes the logged details in various scenarios:

Scenario	ECS option set in the DNS Query	ECS option set in the DNS Response	Logged Details
Both query logging and extended logging enabled	Yes	Yes	ECS option is logged with the string "ECS/R/" and the Scope Prefix-Length is set to the calculated value.
Both query logging and extended logging enabled	Yes	No	ECS option is logged with the string "ECS/Q" and the Scope Prefix-Length is set to zero.
Query logging is enabled, but extended logging is not enabled	Yes	Yes	ECS option is logged with the string "ECS/Q/" and the Scope Prefix-Length is set to zero.

Scenario	ECS option set in the DNS Query	ECS option set in the DNS Response	Logged Details
Query logging & extended logging are not enabled	Yes	Yes	ECS option is not logged.
Query logging is enabled, but extended logging is not enabled	Yes	No	ECS option is logged with the string “ECS/Q/” and the Scope Prefix-Length is set to zero.
Query logging is not enabled, but extended logging is enabled	Yes	Yes	ECS option is logged with the string “ECS/R/” and the Scope Prefix-Length is set to the calculated value.
Query logging is not enabled, but extended logging is enabled	Yes	No	ECS option is not logged.

Understand the record logging format

Following is an example of the record logging format in a Syslog message:

```
<domainname>/<record type>/ <record ttl> / <resource record data>#<resource record data>#......##
<!--NeedCopy-->
```

where:

Record Type	Sample Format	Resource Record Data / Format
Address (A) record	A/5/1.1.1.1#1.1.1.2#1.1.1.3##	IPv4 address
AAAA record	AAAA/5/1::1#1::2#1::3##	IPv6 address
SOA record	SOA/3600/ns1.dnslogging.test./ 3/3600/5##	Origin server, contact, and other details. Resource record format is :

Record Type	Sample Format	Resource Record Data / Format
< originServer >/<contact>/<serial number>/<refresh rate>/<retry>/<expire>/<minimum>##		
NS record	NS/5/ns1.dnslogging.test	Host name of the nameserver.
MX record	#MX/5/10/host1.dnslogging.test.#11/host2.dnslogging.test.##	Preference followed by mail exchange server host name
CNAME record logging	CNAME/5/host1.dnslogging.test	Canonical name
SRV record	SRV/5/1/2/3/host1.dnslogging.test.#4/5/6/host2.dnslogging.test.##	Resource record format: <priority>/<weight>/<port>/<target>#
TXT record	TXT/5/dns+logging##	Data comprises all the texts.
NAPTR record	NAPTR/5/10/11////dnslogging#20/21/R/SIP//sip.dnslogging.test##	Resource record format: <order>/<preference>/<flags>/<services>/<re expression>/<replacement string>#
DNSKEY record	DNSKEY/5/1/3/5/AwEAAanP0K+ JZgiDBZhSON	Resource record format: <flags>/<protocol>/<algorithm>/<public key in base64 encoding>#
PTR record	PTR/3600/test.com.#test4.com.##	Domain name

Limitations of DNS logging

DNS logging has the following limitations:

- If response logging is enabled, only the following record types are logged:
    - Address (A) record
    - AAAA record
    - SOA record
    - NS record
    - MX record
    - CNAME record
    - SRV record
    - TXT record
    - NAPTR record
    - DNSKEY record

  - PTR record

  For all other record types, only L3/L4 parameters, DNS Header, and Question section are logged.

- RRSIG records are not logged even if response logging is enabled.
- DNS64 is not supported.
- DNS proactive update requests or responses are logged according to the settings in the default profile.
- On the virtual server, if sessionless option and response logging is enabled, L3/L4 parameters, DNS Header, and DNS Question section are logged instead of the response.
- The maximum size of the syslog message is 1024 bytes.
- If you have set DNS profile for a DNS policy with action type Rewrite Response, NetScaler appliance does not log the query or the manipulated responses. To log the required information you need to use audit message action in the DNS policy.
- DNS transactions that are due to DNS monitoring traffic are not logged.

Configuring DNS logging

Following is an overview of configuring DNS logging:

1. Create a Syslog action and enable DNS in the action.
2. Create a Syslog policy and specify the Syslog action in the policy.
3. Globally bind the Syslog policy to enable logging of all NetScaler system events. Or, bind the Syslog policy to a specific load balancing virtual server.
4. Create a DNS profile and define any of the following type of logging that you want to enable:
   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging
5. Configure any of the following, based on your requirement:
   - DNS service and virtual server for DNS
   - ADNS service
   - NetScaler as a forwarder
   - NetScaler as a resolver
6. Set the created DNS profile to one of the DNS entities.

Configure DNS logging for NetScaler configured as DNS Proxy by using the CLI

1. Add a syslog action and enable DNS in the action. At the command prompt, type:

```
add audit syslogAction <name> (<serverIP> | -lbVserverName <string>) [-serverPort <port>] -logLevel <logLevel> ... [-dateFormat <dateFormat>] [-logFacility <logFacility>] [-tcp ( NONE | ALL )] [-acl ( ENABLED | DISABLED )] [-timeZone ( GMT_TIME | LOCAL_TIME )] [-userDefinedAuditlog ( YES | NO )] [-appflowExport ( ENABLED |DISABLED )] [-lsn ( ENABLED | DISABLED )] [-alg ( ENABLED | DISABLED )] [-transport ( TCP | UDP )] [-tcpProfileName <string>] [-maxLogDataSizeToHold <positive_integer>] [-dns ( ENABLED | DISABLED)]
<!--NeedCopy-->
```

Example:

```
add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME -dns ENABLED
```

2. Create a syslog policy and specify the created syslog action in the policy. At the command prompt, type:

   ```
   add audit syslogPolicy <name> <rule> <action>
   ```

   Example:

   ```
   add audit syslogPolicy syslogpol1 ns_true nssyslogact1
   ```

3. Bind the syslog policy globally. At the command prompt, type:

   ```
   bind system global [<policyName> [-priority <positive_integer>]]
   ```

   Example:

   ```
   bind system global syslogpol1
   ```

4. Create a DNS profile and enable any of the following type of logs that you want to configure:

   - DNS Query Logging
   - DNS Answer Section Logging
   - DNS Extended Logging
   - DNS Error Logging

   At the command prompt, type:

   ```
   add dns profile <dnsProfileName> [-dnsQueryLogging ( ENABLED | DISABLED )] [-dnsAnswerSecLogging ( ENABLED | DISABLED )] [-dnsExtendedLogging (ENABLED | DISABLED )] [-dnsErrorLogging ( ENABLED | DISABLED )] [-cacheRecords ( ENABLED | DISABLED )] [-cacheNegativeResponses ( ENABLED | DISABLED )]
   ```

   Example:

   ```
   add dns profile dnsprofile1 -dnsQueryLogging ENABLED
   ```

5. Configure service of type DNS. At the command prompt, type:

   ```
   add service <name> <serverName> <serviceType> <port>
   ```

   Example:

   ```
   add service svc1 10.102.84.140 dns 53
   ```

6. Configure a load balancing virtual server of service type DNS.

   ```
   add lb vserver <name> <serviceType> <ip> <port>
   ```

   Example:

   ```
   add lb vserver lb1 dns 100.100.100.10 53
   ```

7. Bind the service to the virtual server. At the command prompt, type:

   ```
   bind lb vserver <name> <serviceName>
   ```

   Example:

   ```
   bind lb vserver lb1 svc1
   ```

8. Set the created DNS profile to the virtual server. At the command prompt, type:

   ```
   set lb vserver <name> [ - dnsProfileName <string>]
   ```

   Example:

   ```
   set lb vserver lb1 -dnsProfileName dnsprofile1
   ```

Sample DNS logging configuration for NetScaler appliance configured as DNS proxy

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel
CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone
LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  - dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
```

```
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

Sample DNS logging configuration for NetScaler appliance configured as ADNS

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add lb vserver lb1 dns 100.100.100.10 53  – dnsProfileName dnsprofile1
Done
> add service svc1 10.102.84.140 dns 53
Done
> bind lb vserver lb1 svc1
Done
<!--NeedCopy-->
```

Sample DNS logging configuration for NetScaler appliance configured as a forwarder

```
> add audit syslogAction nssyslogact1 10.102.151.136 -logLevel CRITICAL
ERROR WARNING NOTICE INFORMATIONAL DEBUG -logFacility LOCAL4 -timeZone LOCAL_TIME
-dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> add dns nameserver 8.8.8.8  – dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

Sample DNS logging configuration for NetScaler appliance configured as a sesolver

```
> add audit syslogAction nssyslogact1 10.102.151.136
-logLevel CRITICAL ERROR WARNING NOTICE INFORMATIONAL DEBUG -
    logFacility LOCAL4
-timeZone LOCAL_TIME -dns ENABLED
Done
> add audit syslogPolicy syslogpol1 ns_true nssyslogact1
Done
> bind system global syslogpol1
Done
> add dns profile dnsprofile1 -dnsqueryLogging ENABLED
Done
> set dns parameter -recursion enABLED
Done
> add nameserver 1.1.1.100 -local dnsProfileName dnsprofile1
Done
<!--NeedCopy-->
```

Configure policy based logging for DNS

Policy based logging enables you to specify a format for log messages. The contents of a log message are defined by using a default syntax expression. When the message action specified in the policy is performed, the NetScaler appliance constructs the log message from the expression and writes the message to the log file. You can configure the appliance to log only when a particular DNS policy evaluates to True.

> **Note**
>
> If you have set a DNS policy with a DNS profile for the request side, NetScaler appliance logs only the query.

To configure policy based logging for a DNS policy, you must first configure an audit message action. For more information about configuring an audit message action, see Configure the NetScaler appliance for audit logging. After configuring the audit message action, specify the message action in a DNS policy.

Configure policy based logging for a DNS policy by using the CLI

At the command prompt, type the following commands to configure policy based logging for a DNS policy and verify the configuration:

```
-  add dns action <actionName> <actionType> [-IPAddress <ip_addr|ipv6_addr> ... | -viewName <string> | -preferredLocList <string> ...] [-TTL <secs>] [-dnsProfileName <string>]
-  set dns policy <name> [<rule>] [-actionName <string>] [-logAction <string>]
-  show dns policy [<name>]
<!--NeedCopy-->
```

Example 1:

In a GSLB deployment, if you want to respond with different IP addresses to the client requests coming from a particular subnet, instead of responding with IP addresses used for general purposes (such as the IP addresses of internal users), you can configure a DNS policy with the action type as DNS view. In this case, you can configure DNS logging on the specified DNS action such that you can log the specific responses.

```
> add dns profile dns_prof1 -dnsqueryLogging enABLED -dnsanswerSecLogging enABLED
 Done
> add dns view dns_view1
 Done
> add dns action dns_act1 viewName -view dns_view1  - dnsprofilename dns_prof1
 Done
> add dns policy dns_pol1 "CLIENT.IP.SRC.APPLY_MASK(255.255.255.0).EQ(100.100.100.0)”
dns_act1
 Done
> bind dns global dns_pol1 100 -gotoPriorityExpression END -type REQ_DEFAULT
 Done
> bind gslb service site_1_svc -viewName dns_view1 123.1.1.1
 Done
> bind gslb service site_5_svc -view dns_view1 132.1.1.1
 Done
<!--NeedCopy-->
```

Note: In the above configuration, if you query for the domain configured on a GSLB virtual server, for example,
sampletest.com, all the internal users of subnet 100.100.100.0/24 are served with the DNS view IP addresses, and the responses are logged. Client requests for other subnets are not logged.

Example 2:

If you want to log only the queries for the domain

example.com, you can create a DNS profile with query logging enabled and set the DNS profile to a DNS action with the action type
NOOP, and then create a DNS policy and set the DNS action. For example:

```
>add dns profile query_logging -dnsqueryLogging ENABLED
Done
>add dns action dns_act1 NOOP -dnsprofileName query_logging
Done
>add dns policy dns_pol1 DNS.REQ.QUESTION.DOMAIN.EQ("example.com")
    dns_act1
Done
<!--NeedCopy-->
```

Configuring DNS suffixes

September 21, 2020

You can configure DNS suffixes that enable the NetScaler appliance to complete non-fully qualified domain names (non-FQDNs) during name resolution. For example, during the process of resolving the domain name abc (which is not fully qualified), if a DNS suffix example.com is configured, the appliance appends the suffix to the domain name (abc.example.com) and resolves it. If DNS suffixes are not configured, the appliance appends a period to the non-FQDNs and resolves the domain name.

Create DNS suffixes

DNS suffixes have significance and are valid only when the NetScaler is configured as an end resolver or forwarder. You can specify a suffix of up to 127 characters.

Note: The order of DNS suffixes is important. The ADC appliance tries the configured suffixes in a serial order and stops when it gets a successful response for a suffix.

Create DNS suffixes by using the CLI

At the command prompt, type the following commands to create a DNS suffix and verify the configuration:

```
-  add dns suffix <dnsSuffix>
-  show dns suffix <dnsSuffix>
<!--NeedCopy-->
```

Example:

```
> add dns suffix example.com
 Done
> show dns suffix example.com
1)       Suffix: example.com
Done

<!--NeedCopy-->
```

To remove a DNS suffix by using the NetScaler command line, at the NetScaler command prompt, type the `rm dns suffix` command and the name of the DNS suffix.

Create DNS suffixes by using the GUI

Navigate to **Traffic Management > DNS > DNS Suffix** and create DNS suffixes.

DNS ANY query

September 21, 2020

An ANY query is a type of DNS query that retrieves all records available for a domain name. The ANY query must be sent to a name server that is authoritative for a domain.

Behavior in ADNS mode

In the ADNS mode, the NetScaler appliance returns the records held in its local cache. If there are no records in the cache, the appliance returns the NXDOMAIN (negative) response.

If the NetScaler can match the domain delegation records, it returns the NS records. Otherwise, it returns the NS records of the root domain.

Behavior in DNS proxy mode

In proxy mode, the NetScaler appliance checks its local cache. If there are no records in the cache, the appliance passes the query to the server.

Behavior for GSLB domains

If a GSLB domain is configured on the NetScaler appliance and an ANY query is sent for the GSLB domain (or GSLB site domain), the appliance returns the IP address of the GSLB service that it selects through the Load Balancing decision. If the multiple IP response (MIR) option is enabled, the IP addresses of all GSLB services are sent.

For the NetScaler to return these records when it responds to the ANY query, all records corresponding to a GSLB domain must be configured on the NetScaler.

> **Note**
>
> If records for a domain are distributed between the NetScaler and a server, only records configured on the NetScaler are returned.

The NetScaler provides the option to configure DNS views and DNS policies. These are used for performing global server load balancing. For more information, see Global Server Load Balancing.

Configure negative caching of DNS records

September 21, 2020

The NetScaler appliance supports caching of negative responses for a domain. A negative response indicates that information about a requested domain does not exist, or that the server cannot provide an answer for the query. The storage of this information is called negative caching. Negative caching helps speed up responses to queries about a domain.

> **Note:**
>
> Negative caching is supported only when the back-end server is configured as an authoritative DNS (ADNS) server for the queried domain.

A negative response can be one of the following:

- NXDOMAIN error message—If a negative response is present in the local cache, the NetScaler returns an error message (NXDOMAIN). If the response is not in the local cache, the query is forwarded to the server, and the server returns an NXDOMAIN error to the NetScaler appliance. The appliance caches the response locally, then returns the error message to the client.
- NODATA error message—If the domain name in query is valid but records of the given type are not available, the appliance sends a NODATA error message.

When negative caching is enabled, the appliance caches the negative response from the DNS server and serves the future requests from the cache only. This helps speed up responses to queries and also to reduce the DNS traffic. Negative caching can be used in all deployments, that is, when a NetScaler appliance is serving as a proxy, as an end resolver, or as a forwarder.

You can enable or disable negative caching using DNS profile, for more information see, DNS profiles. By default, negative caching is enabled in the default DNS profile (*default-dns-profile*) that are bound by default to a DNS virtual server or in the newly created DNS profile.

Enable or disable negative caching by using the CLI

At the command prompt, type the following commands to enable or disable negative caching and verify the configuration:

```
-  add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED
    )] [-cacheNegativeResponses (ENABLED | DISABLED )]
-  show dns profile [<dnsProfileName>]
<!--NeedCopy-->
```

Example of a default DNS profile:

```
> sh dns profile default-dns-profile
    1)   default-dns-profile
        Query logging : DISABLED        Answer section logging :
            DISABLED
        Extended logging : DISABLED     Error logging : DISABLED
        Cache Records : ENABLED     Cache Negative Responses: ENABLED
Done
<!--NeedCopy-->
```

Example of a newly created DNS profile:

```
> add dnsprofile dns_profile1 -cacheRecords ENABLED -
    cacheNegativeResponses ENABLED
Done
> show dns profile dns_profile1
    1)   dns_profile1
        Query logging : DISABLED        Answer section logging :
            DISABLED
        Extended logging : DISABLED     Error logging : DISABLED
        Cache Records : ENABLED     Cache Negative Responses: ENABLED
Done
<!--NeedCopy-->
```

Specify service or virtual server level DNS parameters by using the CLI

At the command prompt, perform the following:

1. Configure the DNS profile.

   `add dns profile <dnsProfileName> [-cacheRecords ( ENABLED | DISABLED )] [-cacheNegativeResponses (ENABLED | DISABLED )]`

2. Bind the DNS profile to the service or virtual server.

To bind the DNS profile to the service:

`set service <name> [-dnsProfileName <string>]`

Example:

```
>set service service1 -dnsProfileName dns_profile1
Done
<!--NeedCopy-->
```

To bind the DNS profile to the virtual server:

`set lb vserver <name> [-dnsProfileName <string>]`

Example:

```
>set lb vserver lbvserver1 -dnsProfileName dns_profile1
Done
<!--NeedCopy-->
```

Specify service or virtual server level DNS parameters by using the GUI

1. Configure the HTTP profile.

   Navigate to **System** > **Profiles**> **DNS Profile**, and create the DNS profile.

2. Bind the HTTP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing**> **Services/Virtual Servers**, and create the DNS profile, which should be bound to the service/virtual server.

Domain name system security extensions

September 21, 2020

DNS Security Extensions (DNSSEC) is an Internet Engineering Task Force (IETF) standard that aims to provide data integrity and data origin authentication in communications between name servers and clients while still transmitting User Datagram Protocol (UDP) responses in clear text. DNSSEC specifies a mechanism that uses asymmetric key cryptography and a set of new resource records that are specific to its implementation.

The DNSSEC specification is described in RFC 4033, "DNS Security Introduction and Requirements," RFC 4034, "Resource Records for the DNS Security Extensions," and RFC 4035, "Protocol Modifications for the DNS Security Extensions." The operational aspects of implementing DNSSEC within DNS are discussed in RFC 4641, "DNSSEC Operational Practices."

You can configure DNSSEC on the NetScaler ADC. You can generate and import keys for signing DNS zones. You can configure DNSSEC for zones for which the NetScaler is authoritative. You can configure the ADC as a DNS proxy server for signed zones hosted on a farm of backend name servers. If the ADC is authoritative for a subset of the records belonging to a zone for which the ADC is configured as a DNS proxy server, you can include the subset of records in the DNSSEC implementation.

Configure DNSSEC

September 21, 2020

Configuring DNSSEC involves enabling DNSSEC on the NetScaler ADC appliance, creating a Zone Signing Key and a Key Signing Key for the zone, adding the two keys to the zone, and then signing the zone with the keys.

The NetScaler appliance does not act as a DNSSEC resolver. DNSSEC on the ADC is supported only in the following deployment scenarios:

1. ADNS—NetScaler is the ADNS and generates the signatures itself.
2. Proxy—NetScaler acts as a DNSSEC proxy. It is assumed that the NetScaler is placed in front of the ADNS/LDNS servers in a trusted mode. The ADC acts only as a proxy caching entity and does not validate any signatures.

Enable and disable DNSSEC

You must enable DNSSEC on the NetScaler for the ADC to respond to DNSSEC-aware clients. By default, DNSSEC is enabled.

You can disable the DNSSEC feature if you do not want the NetScaler to respond to clients with DNSSEC-specific information.

Enable or disable DNSSEC by using the CLI

At the command prompt, type the following commands to enable or disable DNSSEC and verify the configuration:

```
-   set dns parameter -dnssec ( ENABLED | DISABLED )
-   show dns parameter
<!--NeedCopy-->
```

Example:

```
> set dns parameter -dnssec ENABLED
 Done
> show dns parameter
        DNS parameters:
        DNS retries: 5
         .
         .
         .
         DNSEC Extension: ENABLED
        Max DNS Pipeline Requests: 255
 Done

<!--NeedCopy-->
```

Enable or disable DNSSEC by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Change DNS settings.
3. In the Configure DNS Parameters dialog box, select or clear the Enable DNSSEC Extension check box.

Create DNS keys for a zone

For each DNS zone that you want to sign, you must create two pairs of asymmetric keys. One pair, called the Zone Signing Key, is used to sign all the resource record sets in the zone. The second pair is called the Key Signing Key and is used to sign only the DNSKEY resource records in the zone.

When the Zone Signing Key and Key Signing Key are created, the suffix .key is automatically appended to the names of the public components of the keys and the suffix .private is automatically appended to the names of their private components.

Additionally, the NetScaler also creates a Delegation Signer (DS) record and appends the suffix .ds to the name of the record. If the parent zone is a signed zone, you must publish the DS record in the parent zone to establish the chain of trust.

When you create a key, the key is stored in the /nsconfig/dns/ directory, but it is not automatically published in the zone. After you create a key by using the create dns key command, you must explicitly publish the key in the zone by using the add dns key command. The process of generating a key has been separated from the process of publishing the key in a zone to enable you to use alternative means to generate keys. For example, you can import keys generated by other key-generation programs (such as bind-keygen) by using Secure File Transfer Protocol (SFTP) and then publish the keys in the zone. For more information about publishing a key in a zone, see Publish a DNS key in a zone.

Perform the steps described in this topic to create a Zone Signing Key and then repeat the steps to create a Key Signing Key. The example that follows the command syntax first creates a Zone Signing Key pair for the zone example.com. The example then uses the command to create a Key Signing Key pair for the zone.

Create a DNS key by using the CLI

At the NetScaler command prompt, type the following command to create a DNS key:

`create dns key -zoneName <string> -keyType <keyType> -algorithm RSASHA1 -keySize <positive_integer> -fileNamePrefix <string>`

Example:

```
> create dns key -zoneName example.com -keyType zsk -algorithm RSASHA1
    -keySize 1024 -fileNamePrefix example.com.zsk.rsasha1.1024
File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /
    nsconfig/dns/example.com.zsk.rsasha1.1024.private (private); /
    nsconfig/dns/example.com.zsk.rsasha1.1024.ds (ds)
This operation may take some time, Please wait...
 Done
> create dns key -zoneName example.com -keyType ksk -algorithm RSASHA1
    -keySize 4096 -fileNamePrefix example.com.ksk.rsasha1.4096
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /
    nsconfig/dns/example.com.ksk.rsasha1.4096.private (private); /
    nsconfig/dns/example.com.ksk.rsasha1.4096.ds (ds)
This operation may take some time, Please wait...
 Done
<!--NeedCopy-->
```

Create a DNS key by using the GUI

1. Navigate to Traffic Management > DNS.
2. In the details area, click **Create DNS Key** and create a DNS key.
   Note: For
   File Name Prefix, if you want to modify the file name prefix of an existing key, click the arrow next to the
   Browse button, click either
   Local or
   Appliance (depending on whether the existing key is stored on your local computer or in the /nsconfig/dns/ directory on the appliance), browse to the location of the key, and then double-click the key. The

File Name Prefix box is populated with only the prefix of the existing key. Modify the prefix accordingly.

Publish a DNS key in a zone

A key (Zone Signing Key or Key Signing Key) is published in a zone by adding the key to the NetScaler. A key must be published in a zone before you sign the zone.

Before you publish a key in a zone, the key must be available in the /nsconfig/dns/ directory. Therefore, if you used other means to generate the key—means other than the create dns key command on the NetScaler (for example, by using the bind-keygen program on another computer)—make sure that the key is added to the /nsconfig/dns/ directory before you publish the key in the zone.

If the key has been generated by another program, you can import the key to your local computer and use the NetScaler configuration utility to add the key to the /nsconfig/dns/ directory. Or, you can use other means to import the key to the directory, such as the Secure File Transfer Protocol (SFTP).

You must use the add dns key command for each public-private key pair that you want to publish in a given zone. If you created a Zone Signing Key pair and a Key Signing Key pair for a zone, use the add dns key command to first publish one of the key pairs in the zone and then repeat the command to publish the other key pair. For each key that you publish in a zone, a DNSKEY resource record is created in the zone.

The example that follows the command syntax first publishes the Zone Signing Key pair (that was created for the example.com zone) in the zone. The example then uses the command to publish the Key Signing Key pair in the zone.

Publish a key in a zone by using the CLI

At the command prompt, type the following command to publish a key in a zone and verify the configuration:

```
-  add dns key <keyName> <publickey> <privatekey> [-expires <
   positive_integer> [<units>]] [-notificationPeriod <positive_integer>
    [<units>]] [-TTL <secs>]
-  show dns zone [<zoneName> | -type <type>]
<!--NeedCopy-->
```

Example:

```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.
   com.zsk.rsasha1.1024.private
 Done
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.
   com.ksk.rsasha1.4096.private
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY
         Domain Name : ns1.example.com
                 Record Types : A
         Domain Name : ns2.example.com
                 Record Types : A
 Done
<!--NeedCopy-->
```

Publish a key in a DNS zone by using the GUI

Navigate to
Traffic Management > DNS > Keys.

Note: For
Public Key and
Private Key, to add a key that is stored on your local computer, click the arrow next to the
Browse button, click
Local, browse to the location of the key, and then double-click the key.

Configure a DNS key

You can configure the parameters of a key that has been published in a zone. You can modify the key's expiry time period, notification period, and time-to-live (TTL) parameters. If you change the expiry time period of a key, the NetScaler automatically re-signs all the resource records in the zone with the key, provided that the zone is currently signed with the particular key.

Configure a key by using the CLI

At the command prompt, type the following command to configure a key and verify the configuration:

```
-  set dns key <keyName> [-expires <positive_integer> [<units>]] [-notificationPeriod <positive_integer> [<units>]] [-TTL <secs>]
-  show dns key [<keyName>]
<!--NeedCopy-->
```

Example:

```
> set dns key example.com.ksk -expires 30 DAYS -notificationPeriod 3
   DAYS -TTL 3600
 Done
> show dns key example.com.ksk
1)       Key Name: example.com.ksk
         Expires: 30 DAYS        Notification: 3 DAYS    TTL: 3600
         Public Key File: example.com.ksk.rsasha1.4096.key
         Private Key File: example.com.ksk.rsasha1.4096.private
 Done
<!--NeedCopy-->
```

Configure a key by using the GUI

1. Navigate to Traffic Management > DNS > Keys.
2. In the details pane, click the key that you want to configure, and then click Open.
3. In the Configure DNS Key dialog box, modify the values of the following parameters as shown:
    - Expires—expires
    - Notification Period—notificationPeriod
    - TTL—TTL
4. Click OK.

Sign and unsign a DNS zone

To secure a DNS zone, you must sign the zone with the keys that have been published in the zone. When you sign a zone, the NetScaler creates a Next Secure (NSEC) resource record for each owner name. Then, it uses the Key Signing Key to sign the DNSKEY resource record set. Finally, it uses the Zone Signing Key to sign all the resource record sets in the zone, including the DNSKEY resource record sets and NSEC resource record sets. Each sign operation results in a signature for the resource record sets in the zone. The signature is captured in a new resource record called the RRSIG resource record.

After you sign a zone, you must save the configuration.

Sign a zone by using the CLI

At the command prompt, type the following command to sign a zone and verify the configuration:

```
-  sign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
-  save config
<!--NeedCopy-->
```

Example:

```
> sign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY RRSIG NSEC
         Domain Name : ns1.example.com
                 Record Types : A RRSIG NSEC
         Domain Name : ns2.example.com
                 Record Types : A RRSIG
         Domain Name : ns2.example.com
                 Record Types : RRSIG NSEC
Done
> save config
 Done
<!--NeedCopy-->
```

Unsign a zone by using the CLI

At the command prompt, type the following command to unsign a zone and verify the configuration:

```
-  unsign dns zone <zoneName> [-keyName <string> ...]
-  show dns zone [<zoneName> | -type (ADNS | PROXY | ALL)]
<!--NeedCopy-->
```

Example:

```
> unsign dns zone example.com -keyName example.com.zsk example.com.ksk
 Done
> show dns zone example.com
         Zone Name : example.com
         Proxy Mode : NO
         Domain Name : example.com
                 Record Types : NS SOA DNSKEY
         Domain Name : ns1.example.com
                 Record Types : A
         Domain Name : ns2.example.com
                 Record Types : A
 Done
<!--NeedCopy-->
```

Sign or unsign a zone by using the GUI

1. Navigate to Traffic Management > DNS > Zones.
2. In the details pane, click the zone that you want to sign, and then click Sign/Unsign.
3. In the Sign/Unsign DNS Zone dialog box, do one of the following:
    - To sign the zone, select the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to sign the zone.
      You can sign the zone with more than one Zone Signing Key or Key Signing Key pair.
    - To unsign the zone, clear the check boxes for the keys (Zone Signing Key and Key Signing Key) with which you want to unsign the zone.
      You can unsign the zone with more than one Zone Signing Key or Key Signing Key pair.
4. Click OK.

View the NSEC records for a given record in a zone

You can view the NSEC records that the NetScaler automatically creates for each owner name in the zone.

View the NSEC record for a given record in a zone by using the CLI

At the command prompt, type the following command to view the NSEC record for a given record in a zone:

`show dns nsecRec [<hostName> | -type (ADNS | PROXY | ALL)]`

Example:

```
> show dns nsecRec example.com
1)      Domain Name : example.com
        Next Nsec Name: ns1.example.com
        Record Types : NS SOA DNSKEY RRSIG NSEC
 Done
<!--NeedCopy-->
```

View the NSEC record for given record in a zone by using the GUI

1. Navigate to Traffic Management > DNS > Records > Next Secure Records.
2. In the details pane, click the name of the record for which you want to view the NSEC record. The NSEC record for the record you select is displayed in the Details area.

Remove a DNS key

You remove a key from the zone in which it is published when the key has expired or if the key has been compromised. When you remove a key from the zone, the zone is automatically unsigned with the key. Removing the key with this command does not remove the key files present in the /nsconfig/dns/ directory. If the key files are no longer needed, they have to be explicitly removed from the directory.

Remove a key from the NetScaler by using the CLI

At the command prompt, type the following command to remove a key and verify the configuration:

```
-  rm dns key <keyName>
-  show dns key <keyName>
<!--NeedCopy-->
```

Example:

```
> rm dns key example.com.zsk
 Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]

<!--NeedCopy-->
```

Remove a key from the NetScaler by using the GUI

1. Navigate to **Traffic Management > DNS > Keys**.
2. In the details pane, click the name of the key that you want to remove from the ADC, and then click Remove.

Configure DNSSEC when the NetScaler is authoritative for a zone

September 21, 2020

When the NetScaler ADC is authoritative for a given zone, all the resource records in the zone are configured on the ADC. To sign the authoritative zone, you must create keys (the Zone Signing Key and the Key Signing Key) for the zone, add the keys to the ADC, and then sign the zone, as described in Create DNS keys for a zone, Publish a DNS key in a zone, and Sign and unsign a DNS zone, respectively.

If any global server load balancing (GSLB) domains configured on the ADC belong to the zone being signed, the GSLB domain names are signed along with the other records that belong to the zone.

After you sign a zone, responses to requests from DNSSEC-aware clients include the RRSIG resource records along with the requested resource records. DNSSEC must be enabled on the ADC. For more information about enabling DNSSEC, see Enable and disable DNSSEC.

Finally, after you configure DNSSEC for the authoritative zone, you must save the NetScaler configuration.

Configure DNSSEC for a zone for which the NetScaler is a DNS proxy server

September 21, 2020

The procedure for signing a zone for which the NetScaler ADC is configured as a DNS proxy server depends on whether or not the ADC owns a subset of the zone information owned by the backend name servers. If it does, the configuration is considered a *partial zone ownership configuration*. If the ADC does not own a subset of the zone information, the NetScaler configuration for managing the backend servers is considered a *zone-less DNS proxy server configuration*. The basic DNSSEC configuration tasks for both NetScaler configurations are the same. However, signing the partial zone on the NetScaler requires some additional configuration steps.

Note: The terms
zone-less proxy server configuration and
partial zone are used only in the context of the NetScaler appliance.

Important: When configured in proxy mode, the ADC does not perform signature verification on DNSSEC responses before updating the cache.

If you configure the ADC as a DNS proxy to load balance DNSSEC aware resolvers (servers), you must set the Recursion Available option while configuring the DNS virtual server. If a DNSSEC query arrives with Checking Disabled (CD) bit set, the query is passed on to the server with the CD bit retained, and the response from the server is not cached. In releases prior to 10.5.e build xx.x, the ADC unset the CD bit before passing it to the server and also cached the server response.

Configure DNSSEC for a zone-less DNS proxy server configuration

For a zone-less DNS proxy server configuration, zone signing must be performed on the backend name servers. On the NetScaler, you configure the ADC as a DNS proxy server for the zone. You create a load balancing virtual server of protocol type DNS, configure services on the ADC to represent the name servers, and then bind the services to the load balancing virtual server. For more information about these configuration tasks, see Configure the NetScaler as a DNS proxy server.

When a client sends the ADC a DNS request with the DNSSEC OK (DO) bit set, the ADC checks its cache for the requested information. If the resource records are not available in its cache, the ADC forwards the request to one of the DNS name servers, and then relays the response from the name server to the client. Additionally, the ADC caches the RRSIG resource records along with the response from the name server. Subsequent requests from DNSSEC-aware clients are served from the cache (including the RRSIG resource records), subject to the time-to-live (TTL) parameter. If a client sends a DNS request without setting the DO bit, the ADC responds with only the requested resource records, and does not include the RRSIG resource records that are specific to DNSSEC.

Configure DNSSEC for a partial zone ownership configuration

In some NetScaler configurations, even though the authority for a zone lies with the backend name servers, a subset of the resource records that belong to the zone might be configured on the NetScaler. The ADC owns (or is authoritative for) only this subset of records. Such a subset of records can be considered to constitute a *partial zone* on the ADC. The ADC owns the partial zone. All other records are owned by the backend name servers.

A typical partial zone configuration on the NetScaler is seen when global server load balancing (GSLB) domains are configured on the ADC, and the GSLB domains are a part of a zone for which the backend name servers are authoritative.

Signing a zone that includes only a partial zone on the ADC involves including the partial zone information in the backend name server zone files, signing the zone on the backend name servers, and then signing the partial zone on the ADC. The same key set must be used to sign the zone on the name servers and the partial zone on the ADC.

Sign the zone on the backend name servers

1. Include the resource records that are contained in the partial zone, in the zone files of the name servers.
2. Create keys and use the keys to sign the zone on the backend name servers.

Sign the partial zone on the NetScaler

1. Create a zone with the name of the zone that is owned by the backend name servers. When configuring the partial zone, set the proxyMode parameter to YES. This zone is the partial zone that contains the resource records owned by the ADC.

   For example, if the name of the zone that is configured on the backend name servers is example.com, you must create a zone named example.com on the ADC, with the proxyMode parameter set to YES. For more information about adding a zone, see Configure a DNS zone.

> **Note**
>
> Do not add SOA and NS records for the zone. These records should not exist on the ADC for a zone for which the ADC is not authoritative.

2. Import the keys (from one of the backend name servers) to the ADC and then add them to the /nsconfig/dns/ directory. For more information about how you can import a key and add it to the ADC, see Publish a DNS key in a zone.
3. Sign the partial zone with the imported keys. When you sign the partial zone with the keys, the ADC generates RRSIG and NSEC records for the resource record sets and individual resource records in the partial zone, respectively. For more information about signing a zone, see sign and unsign a DNS zone.

Configure DNSSEC for GSLB domain names

September 21, 2020

If global server load balancing (GSLB) is configured on the NetScaler ADC and the ADC is authoritative for the zone to which the GSLB domain names belong, all GLSB domain names are signed when the zone is signed. For more information about signing a zone for which the ADC is authoritative, see Configure DNSSEC when the NetScaler appliance is authoritative for a zone.

If the GSLB domains belong to a zone for which the backend name servers are authoritative, you must first sign the zone on the name servers, and then sign the partial zone on the ADC to complete the DNSSEC configuration for the zone. For more information, see Configure DNSSEC for a partial zone ownership configuration.

Zone Maintenance

September 21, 2020

From a DNSSEC perspective, zone maintenance involves rolling over zone signing keys and key signing keys when key expiry is imminent. These zone maintenance tasks must be performed manually. The zone is re-signed automatically and does not require any manual intervention.

Re-sign an updated zone

When a zone is updated (add a record or modify an existing record), the appliance automatically re-signs the new (or modified) record. If a zone contains multiple zone signing keys, the appliance re-signs the new (or modified) record with the key used to sign the zone.

Roll over DNSSEC keys

Note: Manually roll over the DNSSEC keys (KSK, ZSK) before they expire.

On the NetScaler, you can use the pre-publish and double signature methods to perform a rollover of the zone signing key and key signing key. More information about these two rollover methods is available in RFC 4641, "DNSSEC Operational Practices."

The following topics map commands on the ADC to the steps in the rollover procedures discussed in RFC 4641.

The key expiry notification is sent through an SNMP trap called dnskeyExpiry. Three MIB variables, dnskeyName, dnskeyTimeToExpire, and dnskeyUnitsOfExpiry are sent along with the dnskeyExpiry SNMP trap. For more information, see *Citrix NetScaler SNMP OID Reference* at NetScaler 12.0 SNMP OID Reference.

Prepublish key rollover

RFC 4641, "DNSSEC Operational Practices" defines four stages for the prepublish key rollover method: initial, new DNSKEY, new RRSIGs, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both key signing keys and zone signing keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example:

  Consider the key, example.com.zsk1, with which the zone example.com is signed. The zone contains only those RRSIGs generated by the example.com.zsk1 key, which is due for expiry. The key signing key is example.com.ksk1.

- **Stage 2: New DNSKEY.** A new key is created and published in the zone (that is, the key is added to the ADC), but the zone is not signed with the new key until the pre-roll phase is complete. In this stage, the zone contains the old key, the new key, and the RRSIGs generated by the old key. Publishing the new key for the complete duration of the pre-roll phase gives the DNSKEY resource record corresponding to the new key time to propagate to the secondary name servers.

  Example:

  A new key example.com.zsk2 is added to the example.com zone. The zone is not signed with example.com.zsk2 until the pre-roll phase is complete. The example.com zone contains DNSKEY resource records for both example.com.zsk1 and example.com.zsk2.

  NetScaler commands:

Perform the following tasks on the ADC:

- Create a DNS key by using the `create dns key` command.

  For more information about creating a DNS key, including an example, see Create DNS keys for a zone.

- Publish the new DNS key in the zone by using the `add dns key` command.

  For more information about publishing the key in the zone, including an example, see Publish a DNS key in a zone.

- **Stage 3: New RRSIGs.** The zone is signed with the new DNS key and then unsigned with the old DNS key. The old DNS key is not removed from the zone and remains published until the RRSIGs generated by the old key expire.

  Example:

  The zone is signed with example.com.zsk2 and then unsigned with example.com.zsk1. The zone continues to publish example.com.zsk1 until the RRSIGs generated by example.com.zsk1 expire.

  NetScaler commands:

  Perform the following tasks on the ADC:

  - Sign the zone with the new DNS key by using the `sign dns zone` command.
  - Unsign the zone with the old DNS key by using the `unsign dns zone` command.

  For more information about signing and unsigning a zone, including examples, see Sign and unsign a DNS zone.

- **Stage 4: DNSKEY Removal.** When the RRSIGs generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example:

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  NetScaler commands

  On the ADC, you remove the old DNS key by using the `rm dns key` command. For more information about removing a key from a zone, including an example, see Remove a DNS key.

Double signature key rollover

RFC 4641, "DNSSEC Operational Practices" defines three stages for double signature key rollover: initial, new DNSKEY, and DNSKEY removal. Each stage is associated with a set of tasks that you must perform on the ADC. Following are the descriptions of each stage and the tasks that you must perform. The rollover procedure described here can be used for both key signing keys and zone signing keys.

- **Stage 1: Initial.** The zone contains only those key sets with which the zone has currently been signed. The state of the zone in the initial stage is the state of the zone just before you begin the key rollover process.

  Example:

  Consider the key, example.com.zsk1, with which the zone example.com is signed. The zone contains only those RRSIGs generated by the example.com.zsk1 key, which is due for expiry. The key signing key is example.com.ksk1.

- **Stage 2: New DNSKEY.** The new key is published in the zone and the zone is signed with the new key. The zone contains the RRSIGs that are generated by the old and the new keys. The minimum duration for which the zone must contain both sets of RRSIGs is the time required for all the RRSIGs to expire.

  Example:

  A new key example.com.zsk2 is added to the example.com zone. The zone is signed with example.com.zsk2. The example.com zone now contains the RRSIGs generated from both keys.

  NetScaler commands

  Perform the following tasks on the ADC:

  - Create a DNS key by using the `create dns key` command.

    For more information about creating a DNS key, including an example, see Create DNS keys for a zone.

  - Publish the new key in the zone by using the `add dns key` command.

    For more information about publishing the key in the zone, including an example, see Publish a DNS key in a zone.

  - Sign the zone with the new key by using the `sign dns zone` command.

    For more information about signing a zone, including examples, see Sign and unsign a DNS zone.

- **Stage 3: DNSKEY Removal.** When the RRSIGs generated by the old DNS key expire, the old DNS key is removed from the zone.

  Example:

  The old DNS key example.com.zsk1 is removed from the example.com zone.

  NetScaler commands:

  On the ADC, you remove the old DNS key by using the `rm dns key` command.

  For more information about removing a key from a zone, including an example, see Remove a DNS key.

Offload DNSSEC operations to the NetScaler

September 21, 2020

For DNS zones for which your DNS servers are authoritative, you can offload DNSSEC operations to the NetScaler. In a DNSSEC offloading deployment, a DNS server sends unsigned responses. The ADC signs the response on the fly before relaying it to the client. The ADC also caches the signed response. Apart from reducing the load on the DNS servers, offloading DNSSEC operations to the ADC gives you the following benefits:

- You can sign records that the DNS servers generate programmatically. Such records cannot be signed by routine zone signing operations performed on the DNS servers.
- You can serve signed responses to clients even if you have not implemented DNSSEC on your servers.

For setting up DNSSEC offloading, you must configure a DNS load balancing virtual server, configure services that represent the DNS servers, and then bind the services to the virtual server. For information about configuring a DNS load balancing virtual server, configuring services, and binding the services to the virtual server, see Configure a DNS zone.

You must create a zone entity on the ADC for each DNS zone whose DNSSEC operations you want to offload. For each DNS zone, you must enable the Proxy Mode and DNSSEC Offload parameters. You can optionally configure NSEC record generation for an offloaded zone. To create a DNS zone entity for DNSSEC offloading, follow the instructions in this topic.

To complete the configuration, you must generate DNS keys for the zone, add the keys to the zone, and then sign the zone with the keys. This process is the same as for normal DNSSEC. For information about creating keys, adding keys to a zone, and signing the zone, see Domain name system security extensions.

After you configure DNS offloading, you must flush the DNS cache on the NetScaler. Flushing the DNS cache ensures that any unsigned records in the cache are removed and subsequently replaced by signed records. For information about flushing the DNS cache, see Flush DNS records.

Enable DNSSEC offloading for a zone by using the CLI

At the command line, type the following commands to enable DNSSEC offloading for a zone and verify the configuration:

```
-  add dns zone <zoneName> -proxyMode YES -dnssecOffload ENABLED [-nsec
     ( ENABLED | DISABLED )
-  show dns zone
<!--NeedCopy-->
```

Example:

```
> add dns zone example.com -proxyMode YES -dnssecOffload ENABLED nsec ENABLED
 Done
> show dns zone example.com
     Zone Name : example.com
     Proxy Mode : YES
     DNSSEC Offload: ENABLED    NSEC: ENABLED
 Done
<!--NeedCopy-->
```

Enable DNSSEC offloading for a zone by using the GUI

1. Navigate to **Traffic Management > DNS > Zones**.
2. In the details pane, do one of the following:
    - To create a zone on the NetScaler, click Add.
    - To configure DNSSEC offloading for an existing zone, double-click the zone.
3. In the Create DNS Zone or Configure DNS Zone dialog box, select the Proxy Mode and DNSSEC Offload check boxes.
4. Optionally, if you want the NetScaler to generate NSEC records for the zone, select the NSEC check box.

Firewall Load Balancing

September 21, 2020

Firewall load balancing distributes traffic across multiple firewalls, providing fault tolerance and increased throughput. Firewall load balancing protects your network by:

- Dividing the load between the firewalls, which eliminates a single point of failure and allows the network to scale.
- Increasing high availability.

Configuring a NetScaler appliance for firewall load balancing is similar to configuring load balancing, with the exception that the recommended service type is ANY, recommended monitor type is PING, and the load balancing virtual server mode is set to MAC.

You can set up firewall load balancing in a sandwich, an enterprise, or multiple-firewall environment configuration. The sandwich environment is used for load balancing traffic entering the network from outside and traffic leaving the network to the internet and involves configuring two NetScaler appliances, one on each side of a set of firewalls. You configure an enterprise environment for load balancing traffic leaving the network to the internet. The enterprise environment involves configuring a single NetScaler appliance between the internal network and the firewalls that provide access to the Internet. The multiple-firewall environment is used for load balance traffic coming from another firewall. Having firewall load balancing enabled on both the sides of NetScaler appliance improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic. The multiple-firewall environment involves configuring a NetScaler appliance sandwiched between two firewalls.

Important: If you configure static routes on the NetScaler appliance for the destination IP address and enable L3 mode, the NetScaler appliance uses its routing table to route the traffic instead of sending the traffic to the load balancing vserver.

Note: For FTP to work, an additional virtual server or service should be configured on the NetScaler appliance with IP address and port as * and 21 respectively, and the service type specified as FTP. In this case, the NetScaler appliance manages the FTP protocol by accepting the FTP control connection, modifying the payload, and managing the data connection, all through the same firewall.

Firewall Load Balancing supports only some of the load balancing methods supported on the NetScaler appliance. Also, you can configure only a few types of persistence and monitors.

Firewall Load Balancing Methods

The following load balancing methods are supported for firewall load balancing.

- Least Connections
- Round Robin
- Least Packets
- Least Bandwidth
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port hash
- Least Response Time Method (LRTM)
- Custom Load

Firewall Persistence

Only SOURCEIP, DESTIP, and SOURCEIPDESTIP based persistence are supported for firewall load balancing.

Firewall Server Monitoring

Only PING and transparent monitors are supported in firewall load balancing. You can bind a PING monitor (default) to the backend service that represents the firewall. If a firewall is configured not to respond to ping packets, you can configure transparent monitors to monitor hosts on the trusted side through individual firewalls.

Sandwich Environment

December 28, 2020

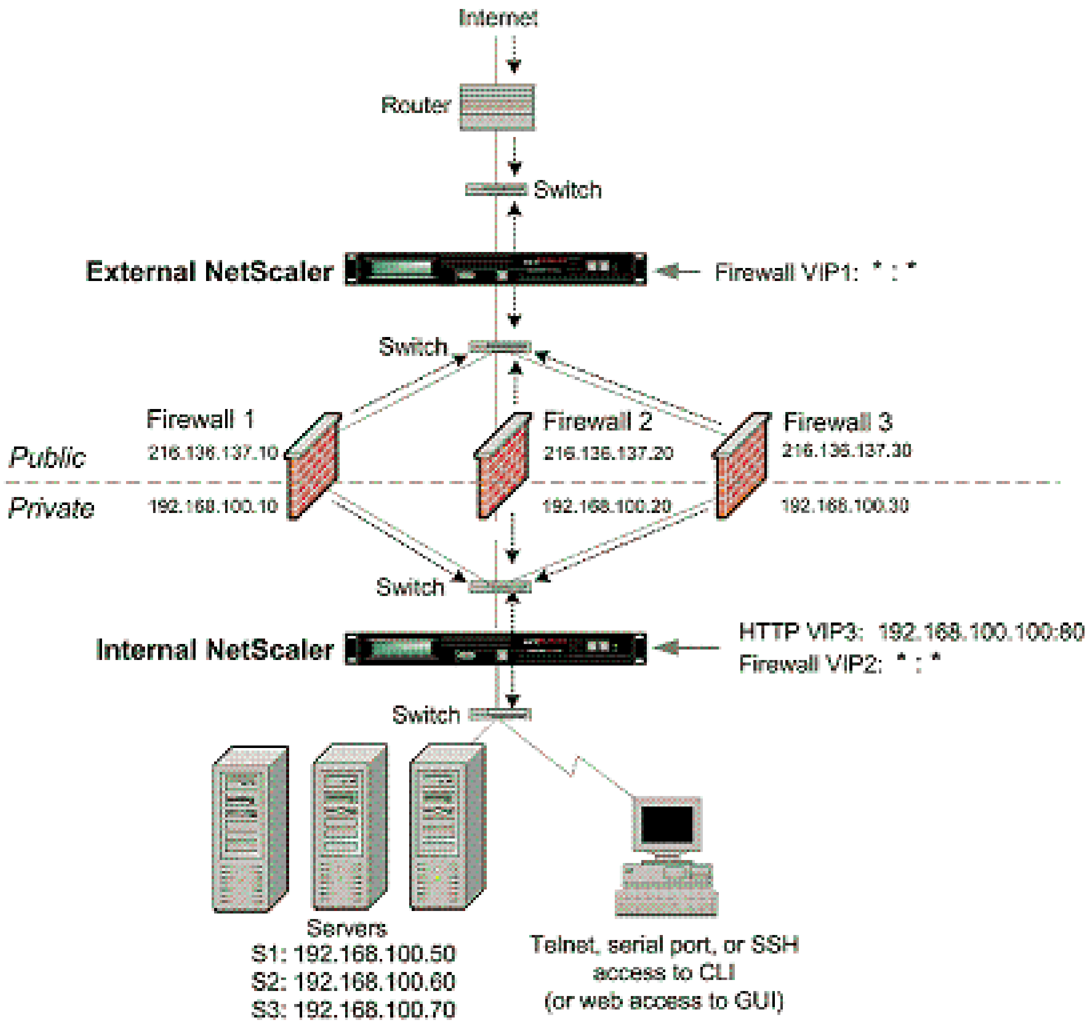
A NetScaler deployment in a sandwich mode is capable of load balancing network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

In this setup, a NetScaler is located on each side of a set of firewalls. The NetScaler placed between the firewalls and the Internet, called the external NetScaler that handles ingress traffic selects the best firewall, based on the configured method. The NetScaler between the firewalls and the private network, called the internal NetScaler tracks the firewall from which the initial packet for a session is received. It then makes sure that all subsequent packets for that session are sent to the same firewall.

The internal NetScaler can be configured as a regular traffic manager to load balance traffic across the private network servers. This configuration also allows traffic originating from the private network (egress) to be load balanced across the firewalls.

The following diagram shows the sandwich firewall load balancing environment.

Figure 1. Firewall Load Balancing (Sandwich)


Internet
Router
Switch
External NetScaler
Firewall VIP1: * : *
Switch
Firewall 1
216.136.137.10
192.168.100.10
Firewall 2
216.136.137.20
192.168.100.20
Firewall 3
216.136.137.30
192.168.100.30
Public
Private
Switch
Internal NetScaler
HTTP VIP3: 192.168.100.100:80
Firewall VIP2: * : *
Switch
Servers
S1: 192.168.100.50
S2: 192.168.100.60
S3: 192.168.100.70
Telnet, serial port, or SSH access to CLI (or web access to GUI)

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the External NetScaler in a Sandwich Environment

Perform the following tasks to configure the external NetScaler in a sandwich environment

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server for traffic coming from the Internet.
- Configure the virtual server in MAC rewrite mode.
- Bind services to the wildcard virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   OFF
 2)     Surge Protection               SP                   ON
 3)     Load Balancing                 LB                   ON
 .
 .
 .
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in **Configure Basic Features**, select **Load Balancing**.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and add a service. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
To bind a PING monitor, type the following command:
bind monitor PING fw-svc1
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the configuration utility

Navigate to
Traffic Management > Load Balancing > Monitors, and then create and bind a transparent monitor.

Configure a wildcard virtual server for traffic coming from the Internet

To configure a wildcard virtual server for traffic coming from the Internet by using the command line interface

At the command prompt, type:

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers and create a wildcard virtual server. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section, and click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

Bind services to the wildcard virtual server

To bind a service to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a service to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to bind the service.
2. Click in the **Services** section and select a service to bind.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
save ns config
show vserver
<!--NeedCopy-->
```

Example:

```
save config
sh lb vserver FWLBVIP1
FWLBVIP1 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 06:40:14 2010
        Time since last state change: 0 days, 00:00:11.240
        Effective State: UP  ARP:DISABLED
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: SRCIPDESTIPHASH
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw_svc_1 (10.102.29.251: *) - ANY State: UP  Weight: 1
2) fw_svc_2 (10.102.29.18: *) - ANY State: UP   Weight: 1
 Done
show service fw-svc1
        fw-svc1 (10.102.29.251:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 10:04:50 2010
        Time since last state change: 0 days, 00:00:38.120
        Server Name: 10.102.29.251
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 5       Failed [Total: 5 Current: 5]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.415 millisec
 Done
<!--NeedCopy-->
```

Configuring the Internal NetScaler in a Sandwich Environment

Perform the following tasks to configure the internal NetScaler in a sandwich environment

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.
- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature LB
show ns feature
<!--NeedCopy-->
```

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   OFF
 2)     Surge Protection               SP                   ON
```

```
3)      Load Balancing                  LB                    ON
.
.
.
24)     NetScaler Push                  push                  OFF
Done
<!--NeedCopy-->
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and add a service. Specify **ANY** in the **Protocol** field and * in the Port field.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors and create a monitor.
2. In the **Create Monitor** dialog box, enter the required parameters and select **Transparent**.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server for traffic coming from the Internet by using the configuration utility

Navigate to
Traffic Management > Load Balancing > Virtual Servers and create a wildcard virtual server. Specify **ANY** in the Protocol field and * in the Port field.

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
    - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1).
2. Edit the **Basic Settings** section and. click **more**.
3. From the **Redirection Mode** drop-down list, select **MAC Based**.

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1), and then click Open.
3. On the Advanced tab, under Redirection Mode, click MAC-Based.
4. Click OK.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

To configure a service for each virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services and configure a service for each virtual server. Specify **HTTP** in the **Protocol** field and select **HTTP** under **Available Monitors**.

To configure a service for each virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
    - Service Name—name
    - Server—serverName
    - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

Navigate to **Traffic Management > Load Balancing > Services**, double-click a service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Services and configure an HTTP virtual server. Specify **HTTP** in the **Protocol** field.

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
    - Name—name
    - IP Address—IPAddress
      Note: If the virtual server uses IPv6, select the
      IPv6 check box and enter the address in IPv6 format (for example,
      1000:0000:0000:0000:0005:0600:700a:888b).
    - Port—port
4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in a Sandwich Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *    80         HTTP           UP      5/s
        0/s
Two                        *     0          TCP         DOWN      0/s
        0/s
Three                      *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done


<!--NeedCopy-->
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

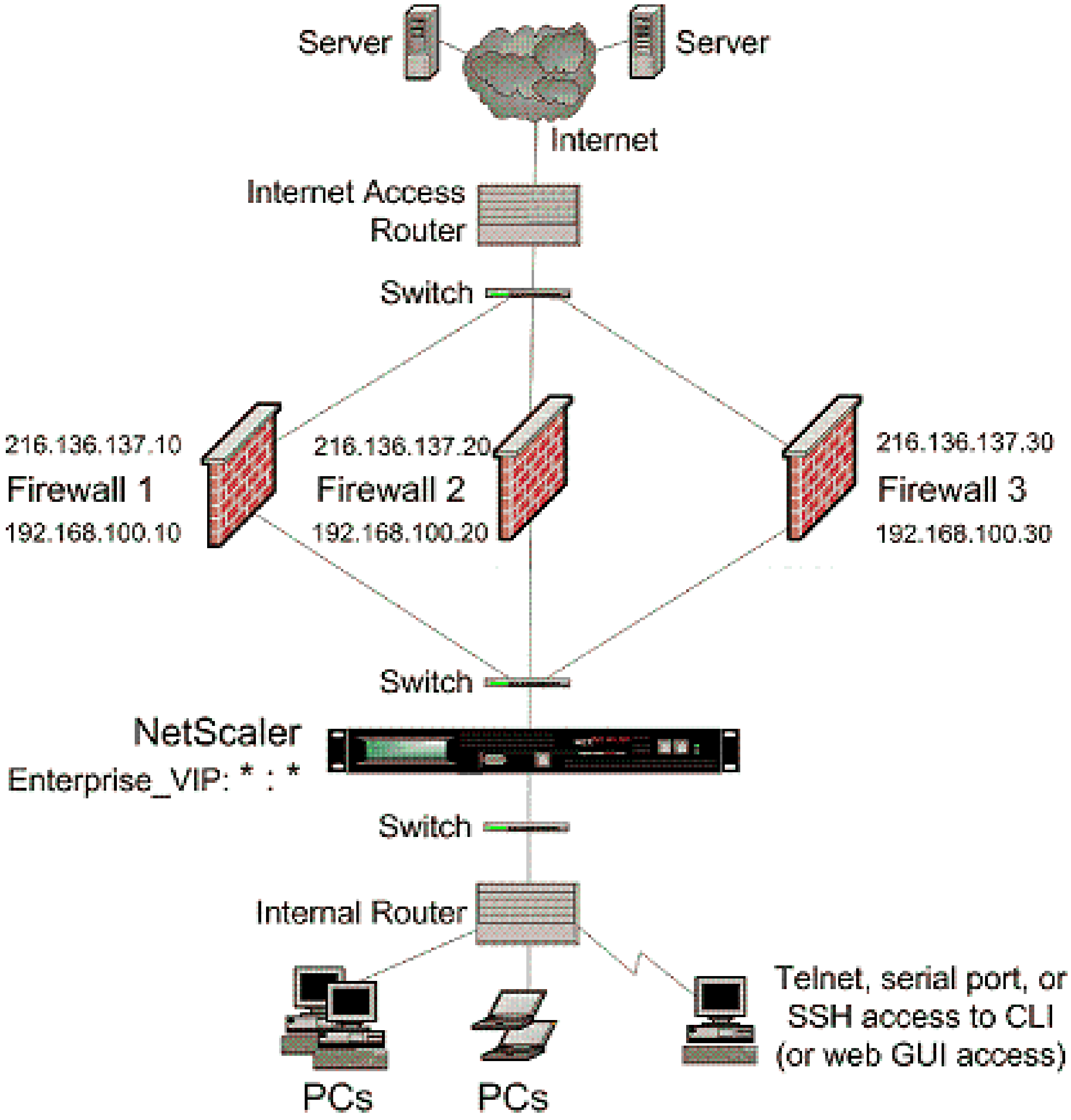
Enterprise Environment

September 21, 2020

In the enterprise setup, the NetScaler is placed between the firewalls connecting to the public Internet and the internal private network and handles egress traffic. The NetScaler selects the best firewall based on the configured load balancing policy.

The following diagram shows the enterprise firewall load balancing environment

Figure 1. Firewall Load Balancing (Enterprise)


Server
Server
Internet
Internet Access Router
Switch
216.136.137.10
Firewall 1
192.168.100.10
216.136.137.20
Firewall 2
192.168.100.20
216.136.137.30
Firewall 3
192.168.100.30
Switch
NetScaler
Enterprise_VIP: * : *
Switch
Internal Router
PCs
PCs
Telnet, serial port, or SSH access to CLI (or web GUI access)

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and vserver with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the NetScaler in an Enterprise Environment

Perform the following tasks to configure a NetScaler in an enterprise environment.

For traffic from the server (egress)

- Enable the load balancing feature.
- Configure a wildcard service for each firewall.
- Configure a monitor for each wildcard service.

- Configure a wildcard virtual server to load balance the traffic sent to the firewalls.
- Configure the virtual server in MAC rewrite mode.
- Bind firewall services to the wildcard virtual server.

For traffic across private network servers

- Configure a service for each virtual server.
- Configure a monitor for each service.
- Configure an HTTP virtual server to balance traffic sent to the servers.
- Bind HTTP services to the HTTP virtual server.
- Save and Verify the Configuration.

Enable the load balancing feature

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example:

```
> enable ns feature LoadBalancing
 Done
> show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  OFF
 2)     Surge Protection                SP                  ON
 3)     Load Balancing                  LB                  ON
 .
 .
 .
 24)    NetScaler Push                  push                OFF
 Done
<!--NeedCopy-->
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Configure a wildcard service for each firewall

To configure a wildcard service for each firewall by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> ANY *
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
    - Service Name—name
    - Server—serverName
4. In Protocol, select ANY, and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each wildcard service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values as shown:
    - Name*
    - Type*—type
    - Destination IP
    - Transparent

   -* A required parameter
4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure a wildcard virtual server to load balance the traffic sent to the firewalls

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the command line interface

At the command prompt, type:

```
add lb vserver <name> ANY * *
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a wildcard virtual server to load balance the traffic sent to the firewalls by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
    - Name—name
4. In Protocol, select ANY, and in IP Address and Port, select *.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configure the virtual server in MAC rewrite mode

To configure the virtual server in MAC rewrite mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB-1), and then click Open.
3. On the Advanced tab, under Redirection Mode, click MAC-Based.
4. Click OK.

Bind firewall services to the wildcard virtual server

To bind firewall services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind firewall services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Configure a service for each virtual server

To configure a service for each virtual server by using the command line interface

At the command prompt, type:

```
add service <name> <serverName> HTTP <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
<!--NeedCopy-->
```

To configure a service for each virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName
   - Port—port
4. In Protocol, specify HTTP. Under Available Monitors, select HTTP.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configure a monitor for each service

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
bind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

Configure an HTTP virtual server to balance traffic sent to the servers

To configure an HTTP virtual server to balance traffic sent to the servers by using the command line interface

At the command prompt, type:

```
add lb vserver <name> HTTP <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To configure an HTTP virtual server to balance traffic sent to the servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In the Create Virtual Server (Load Balancing) dialog box, specify values for the following parameters as shown:
    - Name—name

- IP Address—IPAddress
  Note: If the virtual server uses IPv6, select the
  IPv6 check box and enter the address in IPv6 format (for example, **1000:0000:0000:0000:0005:0600:700a:888b**).
- Port—port

4. Under Protocol, select HTTP.
5. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Bind HTTP services to the HTTP virtual server

To bind HTTP services to the wildcard virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind HTTP services to the wildcard virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and select a virtual server.
2. Click in the Service section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Save and Verify the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection
```

```
                    establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

To save and verify the configuration by using the configuration utility

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in an Enterprise Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler appliance, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *     80        HTTP           UP      5/s
        0/s
Two                        *      0         TCP         DOWN      0/s
        0/s
Three                      *   2598         TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done


<!--NeedCopy-->
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

Updated: 2013-08-28

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

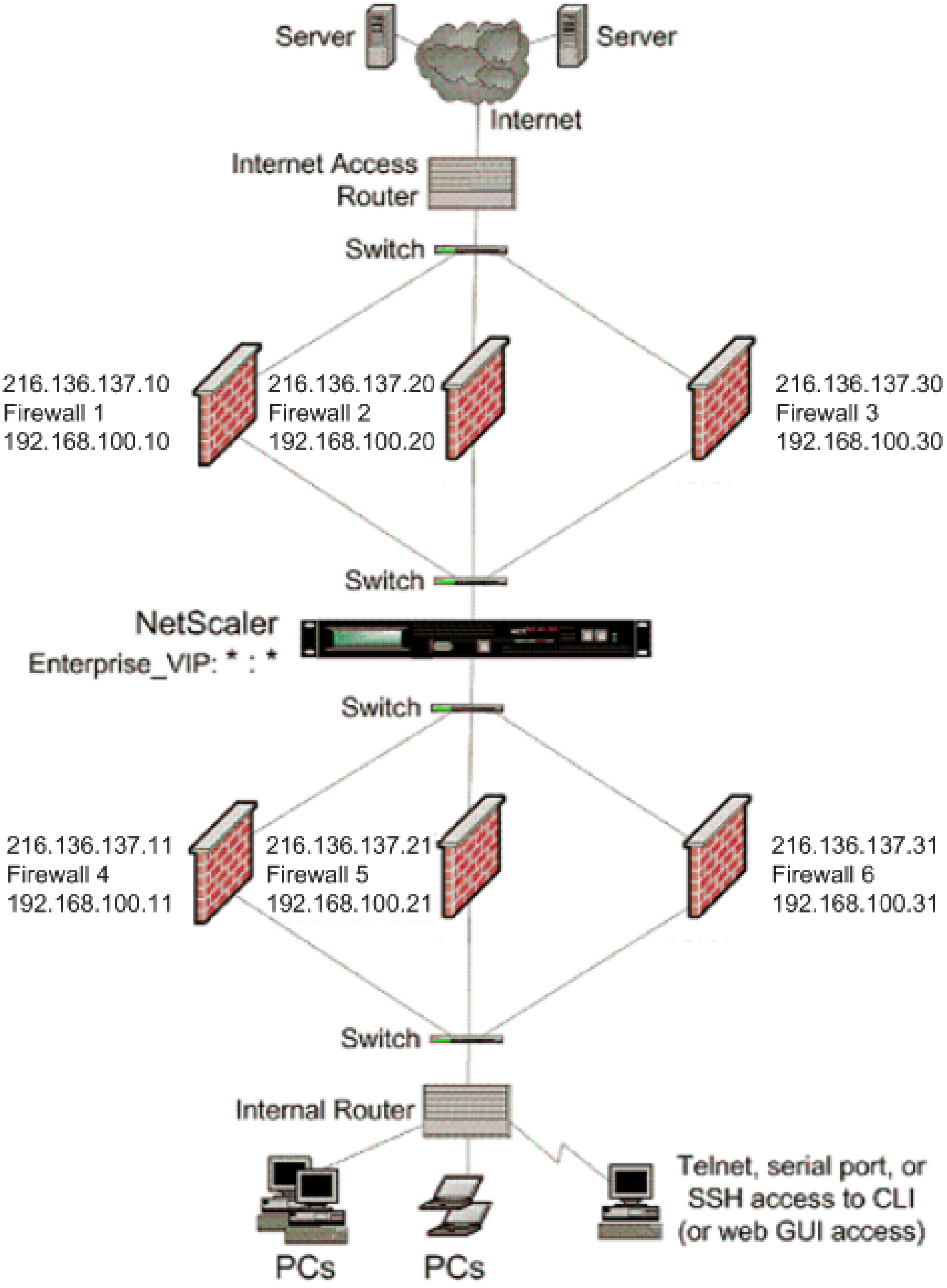
Multiple-Firewall Environment

September 21, 2020

In a multiple-firewall environment, the NetScaler appliance is placed between two sets of firewalls, the external set connecting to the public Internet, and the internal set connecting to the internal private network. The external set typically handles the egress traffic. These firewalls mainly implement access control lists to allow or deny access to external resources. The internal set typically handles the ingress traffic. These firewalls implement security to safeguard the intranet from malicious attacks apart from load-balancing the ingress traffic. The multiple-firewall environment allows you to load-balance traffic coming from another firewall. By default, the traffic coming from a firewall is not load balanced on the other firewall across a NetScaler appliance. Having firewall load balancing enabled on both the sides of NetScaler improves the traffic flow in both the egress and ingress direction and ensures faster processing of the traffic.

The following figure shows a multiple-firewall load balancing environment

Figure 1. Firewall Load Balancing (multiple-firewall)


Server
Server
Internet
Internet Access Router
Switch
216.136.137.10
Firewall 1
192.168.100.10
216.136.137.20
Firewall 2
192.168.100.20
216.136.137.30
Firewall 3
192.168.100.30
Switch
NetScaler
Enterprise_VIP: * : *
Switch
216.136.137.11
Firewall 4
192.168.100.11
216.136.137.21
Firewall 5
192.168.100.21
216.136.137.31
Firewall 6
192.168.100.31
Switch
Internal Router
Telnet, serial port, or SSH access to CLI (or web GUI access)
PCs
PCs

With a configuration like the one shown in Figure 1, you can configure the NetScaler to load balance the traffic through the an internal firewall even if it is load balanced by an external firewall. For example, with this feature configured, the traffic coming from the external firewalls (firewalls 1, 2, and 3) is load balanced on the internal firewalls (firewalls 4, 5, and 6) and vice versa.

Firewall load balancing is supported only for MAC mode LB virtual server.

The service type ANY configures the NetScaler to accept all traffic.

To avail benefits related to HTTP and TCP, configure the service and virtual server with type HTTP or TCP. For FTP to work, configure the service with type FTP.

Configuring the NetScaler in a Multiple-Firewall Environment

To configure a NetScaler appliance in a multiple-firewall environment, you have to enable the load balancing feature, configure a virtual server to load balance the egress traffic across the external firewalls, configure a virtual server to load balance the ingress traffic across the internal firewalls, and enable firewall load balancing on the NetScaler appliance. To configure a virtual server to load balance traffic across a firewall in the multiple-firewall environment, you need to:

1. Configure a wildcard service for each firewall
2. Configure a monitor for each wildcard service
3. Configure a wildcard virtual server to load balance the traffic sent to the firewalls
4. Configure the virtual server in MAC rewrite mode
5. Bind firewall services to the wildcard virtual server

Enabling the load balancing feature

To configure and implement load balancing entities such as services and virtual servers, you need to enable the load balancing feature on the NetScaler device.

To enable load balancing by using the CLI:

At the command prompt, type the following command to enable load balancing and verify the configuration:

```
enable ns feature <featureName>
show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature LoadBalancing
Done
show ns feature
Feature Acronym Status
------- ------- ------
```

```
1) Web Logging WL OFF
2) Surge Protection SP ON
3) Load Balancing LB ON
.
.
.
24) NetScaler Push push OFF
Done
<!--NeedCopy-->
```

To enable load balancing by using the GUI:

1. In the navigation pane, expand System, and then click Settings.
2. In the Settings pane, under Modes and Features, click Change basic features.
3. In the Configure Basic Features dialog box, select the Load Balancing check box, and then click Ok.

Configuring a wildcard service for each firewall

To accept traffic from all the protocols, you need to configure wildcard service for each firewall by specifying support for all the protocols and ports.

To configure a wildcard service for each firewall by using the CLI:

At the command prompt, type the following command to configure support for all the protocols and ports:

```
add service <name>@ <serverName> <serviceType> <port_number>
<!--NeedCopy-->
```

Example:

```
add service fw-svc1 10.102.29.5 ANY *
<!--NeedCopy-->
```

To configure a wildcard service for each firewall by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Services dialog box, specify values for the following parameters as shown:
   - Service Name—name
   - Server—serverName

   -* A required parameter

4. In Protocol, select Any and in Port, select *.
5. Click Create, and then click Close. The service you created appears in the Services pane.

Configuring a monitor for each service

A PING monitor is bound by default to the service. You will need to configure a transparent monitor to monitor hosts on the trusted side through individual firewalls. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the firewall is UP but one of the next hop devices from that firewall is down, the appliance includes the firewall while performing load balancing and forwards the packet to the firewall. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the firewall) are down, the service is marked as DOWN and the firewall is not included when the appliance performs firewall load balancing.

Binding a transparent monitor will override the PING monitor. To configure a PING monitor in addition to a transparent monitor, after you create and bind a transparent monitor, you need to bind a PING monitor to the service.

To configure a transparent monitor by using the CLI:

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
   transparent (YES | NO )]
bind lb monitor <monitorName> <serviceName>
<!--NeedCopy-->
```

Example:

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
<!--NeedCopy-->
```

The Citrix ADC appliance learns the server L2 parameters from the monitor that is bound to the service. For UDP-ECV monitors, configure a receive string to enable the appliance to learn the L2 parameters of the server. If the receive string is not configured and the server does not respond, then the appliance does not learn the L2 parameters but the service is set to UP. The traffic for this service is blackholed.

To configure a receive string by using the CLI:

At the command prompt, type the following command:

```
add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-
    transparent (YES | NO )] [-send <string>] [-recv <string>]
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-udp-1 udp-ecv -destip 10.10.10.11 -transparent
    YES - send “test message” - recv “site_is_up”
<!--NeedCopy-->
```

To create and bind a transparent monitor by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the details pane, click Add.
3. In the Create Monitor dialog box, specify values for the following parameters as shown:
    - Name*
    - Type*—type
    - Destination IP
    - Transparent

   -* A required parameter
4. Click Create, and then click Close. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configuring a virtual server to load balance the traffic sent to the firewalls

To load balance any kind of traffic, you need to configure a wildcard virtual server specifying the protocol and port as any value.

To configure a virtual server to load balance the traffic sent to the firewalls by using the CLI:

At the command prompt, type the following command:

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 ANY * *
<!--NeedCopy-->
```

To configure a virtual server to load balance the traffic sent to the firewalls by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, click Add.
3. In Protocol, select Any, and in IP Address and Port, select *.
4. Click Create, and then click Close. The virtual server you created appears in the Load Balancing Virtual Servers pane.

Configuring the virtual server to MAC rewrite mode

To configure the virtual server to use MAC address for forwarding the incoming traffic, you need to enable the MAC rewrite mode.

To configure the virtual server in MAC rewrite mode by using the CLI:

At the command prompt, type the following command:

```
set lb vserver <name>@ -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

To configure the virtual server in MAC rewrite mode by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. On the Advanced tab, under the Redirection Mode mode, click Open.
4. Click Ok.

Binding firewall services to the virtual server

To access a service on NetScaler appliance, you need to bind it to a wildcard virtual server.

To bind firewall services to the virtual server by using the CLI:

At the command prompt, type the following command:

```
bind lb vserver <name>@ <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind firewall services to the virtual server by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Vserver-LB1), and then click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, on the Services tab, select the Active check box next to the service that you want to bind to the virtual server(for example, Service-HTTP-1).
4. Click Ok.

Configuring the multiple-firewall load balancing on the NetScaler appliance

To load balance traffic on both the sides of a NetScaler using firewall load balancing, you need to enable mulitpl-firewall load balancing by using the vServerSpecificMac parameter.

To configure multiple-firewall load balancing by using the CLI:

At the command prompt, type the following command:

```
set lb parameter -vServerSpecificMac <status>
<!--NeedCopy-->
```

Example:

```
set lb parameter -vServerSpecificMac ENABLED
<!--NeedCopy-->
```

To configure multiple-firewall load balancing by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server for which you want to configure the redirection mode (for example, Configure Load Balancing parameters).
3. In the Set Load Balancing Parameters dialog box, select the Virtual Server Specific MAC check box.
4. Click Ok.

Saving and Verifying the Configuration

When you've finished the configuration tasks, be sure to save the configuration. You should also check to make sure that the settings are correct.

To save and verify the configuration by using the CLI:

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- save ns config
- show vserver

Example:

```
save config
show lb vserver FWLBVIP2
        FWLBVIP2 (*:*) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Mon Jun 14 07:22:54 2010
        Time since last state change: 0 days, 00:00:32.760
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  2 (Total)       2 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: A new service is bound
        Mode: MAC
        Persistence: NONE
        Connection Failover: DISABLED

1) fw-int-svc1 (10.102.29.5: *) - ANY State: UP Weight: 1
2) fw-int-svc2 (10.102.29.9: *) - ANY State: UP Weight: 1
 Done
show service fw-int-svc1
        fw-int-svc1 (10.102.29.5:*) - ANY
        State: DOWN
        Last state change was at Thu Jul  8 14:44:51 2010
        Time since last state change: 0 days, 00:01:50.240
        Server Name: 10.102.29.5
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

```

```
1)      Monitor Name: monitor-HTTP-1
                State: DOWN     Weight: 1
                Probes: 9       Failed [Total: 9 Current: 9]
                Last response: Failure - Time out during TCP connection establishment stage
                Response Time: 2000.0 millisec
2)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 3       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.275 millisec
 Done
<!--NeedCopy-->
```

**To save and verify the configuration by using the GUI:

1. In the details pane, click Save.
2. In the Save Config dialog box, click Yes.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers.
4. In the details pane, select the virtual server that you created in step 5 and verify that the settings displayed in the Details pane are correct.
5. Navigate to Traffic Management > Load Balancing > Services.
6. In the details pane, select the service that you created in step 5 and verify that the settings displayed in the Details pane are correct.

Monitoring a Firewall Load Balancing Setup in a Multiple-Firewall Environment

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received

- Rate of hits

To display virtual server statistics by using the CLI:

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler appliance, or for a single virtual server, at the command prompt, type:

```
stat lb vserver [-detail] [<name>]
<!--NeedCopy-->
```

Example:

```
>stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *    80         HTTP           UP      5/s
        0/s
Two                        *     0          TCP         DOWN      0/s
        0/s
Three                      *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done
<!--NeedCopy-->
```

To display virtual server statistics by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the CLI:

At the command prompt, type:

```
stat service <name>
```

```
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

Global Server Load Balancing

November 9, 2020

> **Note**
>
> The GSLB feature is included with the NetScaler Advance and Premium edition licenses. The NetScaler option license is supported with the Standard edition.

NetScaler appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in case of an outage.

In a typical configuration, a local DNS server sends client requests to a GSLB virtual server, to which are bound GSLB services. A GSLB service identifies a load balancing or content switching virtual server, which can be at the local site or a remote site. If the GSLB virtual server selects a load balancing or content switching virtual server at a remote site, it sends the virtual server's IP address to the DNS server, which sends it to the client. The client then resends the request to the new virtual server at the new IP.

The GSLB entities that you must configure are the GSLB sites, the GSLB services, the GSLB virtual servers, load balancing or content switching virtual servers, and authoritative DNS (ADNS) services. You must also configure MEP. You can also configure DNS views to expose different parts of your network to clients accessing the network from different locations.

> **Note**
>
> To take full advantage of the NetScaler GSLB features, you should use NetScaler appliances for load balancing or content switching at each data center, so that your GSLB configuration can use the proprietary Metric Exchange Protocol (MEP) to exchange site metrics.

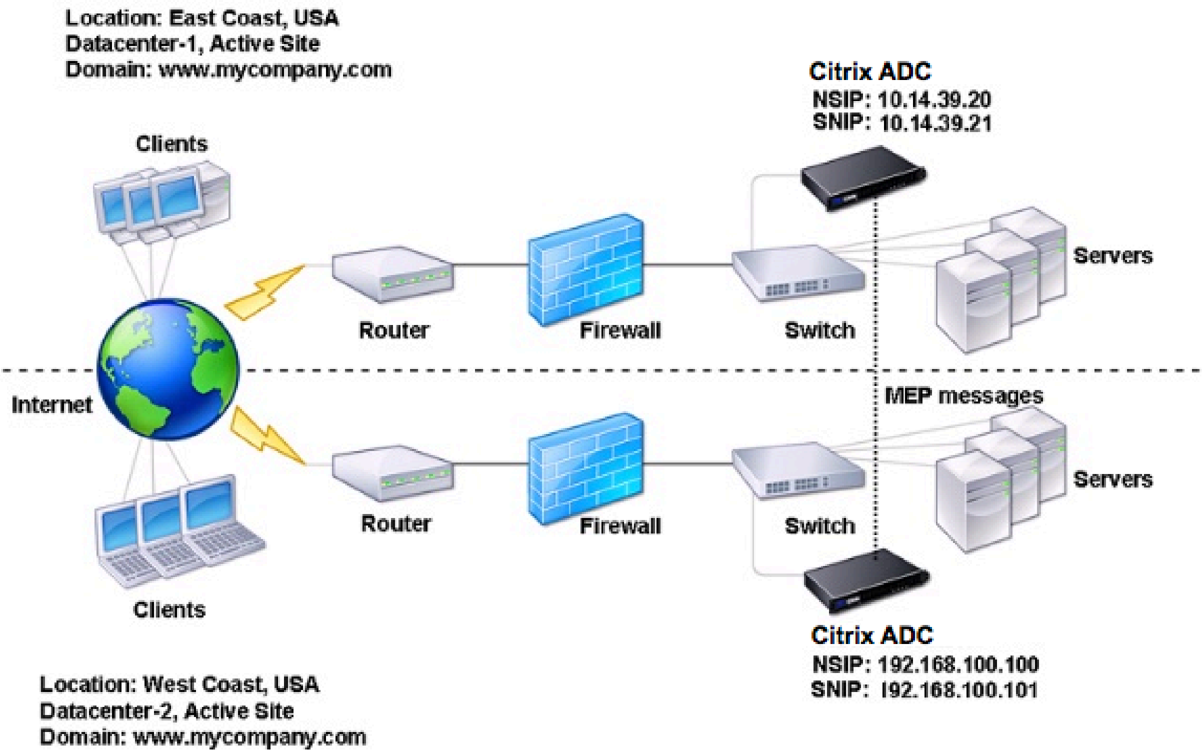
How GSLB Works

September 21, 2020

With ordinary DNS, when a client sends a domain name system (DNS) request, it receives a list of IP addresses of the domain or service. Generally, the client chooses the first IP address in the list and initiates a connection with that server. The DNS server uses a technique called DNS round robin to rotate through the IPs on the list, sending the first IP address to the end of the list and promoting the others after it responds to each DNS request. This technique ensures equal distribution of the load, but it does not support disaster recovery, load balancing based on load or proximity of servers, or persistence.

When you configure GSLB on NetScaler appliances and enable Metric Exchange Protocol (MEP), the appliances use the DNS infrastructure to connect the client to the data center that best meets the criteria that you set. The criteria can designate the least loaded data center, the closest data center, the data center that responds most quickly to requests from the client's location, a combination of those metrics, and SNMP metrics. An appliance keeps track of the location, performance, load, and availability of each data center and uses these factors to select the data center to which to send a client request.

The following figure illustrates a basic GSLB topology.


Location: East Coast, USA
Datacenter-1, Active Site
Domain: www.mycompany.com
Citrix ADC
NSIP: 10.14.39.20
SNIP: 10.14.39.21
Clients
Servers
Router
Firewall
Switch
Internet
MEP messages
Servers
Router
Firewall
Switch
Clients
Citrix ADC
NSIP: 192.168.100.100
SNIP: 192.168.100.101
Location: West Coast, USA
Datacenter-2, Active Site
Domain: www.mycompany.com

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include GSLB sites, GSLB services, GSLB virtual servers, load balancing and/or content switching servers, and ADNS services.

GSLB Deployment Types

September 21, 2020

NetScaler appliances configured for global server load balancing (GSLB) provide for disaster recovery and ensure continuous availability of applications by protecting against points of failure in a wide area network (WAN). GSLB can balance the load across data centers by directing client requests to the closest or best performing data center, or to surviving data centers in the event of an outage.

The following are some of the typical GSLB deployment types:

- Active-active site deployment
- Active-passive site deployment
- Parent-child topology deployment using the MEP protocol

Active-Active Site Deployment

September 21, 2020

An active-active site consists of multiple active data centers. Client requests are load balanced across active data centers. This deployment type can be used when you have a need for global distribution of traffic in a distributed environment.

All the sites in an active-active deployment are active, and all the services for a particular application/domain are bound to the same GSLB vserver. Sites exchange metrics through the Metrics Exchange Protocol (MEP). Site metrics exchanged between the sites include the status of each load balancing and content switching virtual server, current number of connections, current packet rate, and current bandwidth usage. The NetScaler appliance needs this information to perform load balancing across the sites.

An active-active deployment can include a maximum of 32 GSLB sites, because MEP cannot synchronize more than 32 sites. No backup sites are configured in this deployment type.

The NetScaler appliance sends client requests to the appropriate GSLB site as determined by the GSLB method specified in the GSLB configuration.

For an active-active deployment, you can configure the following GSLB methods.

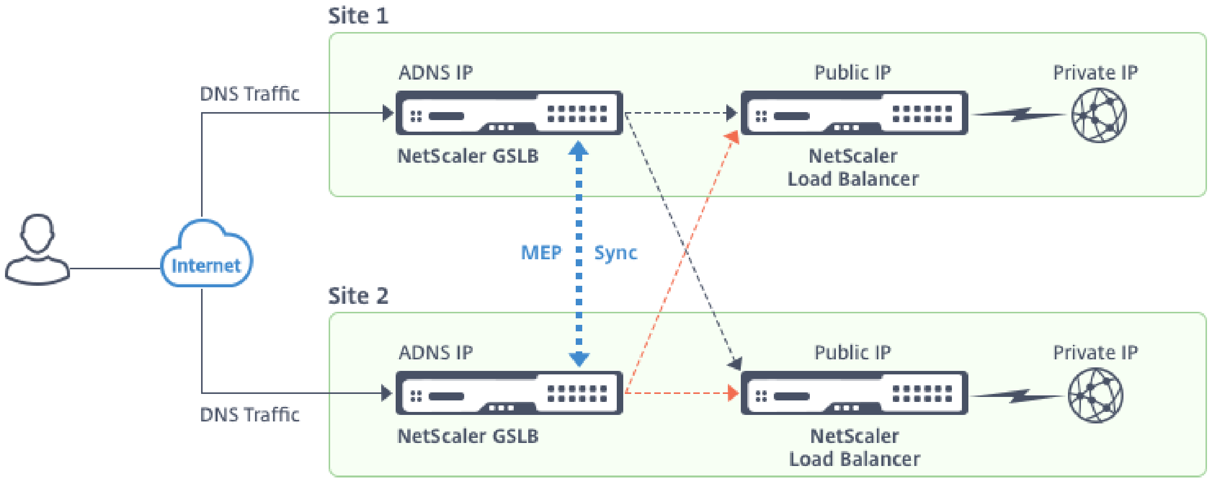
- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

Note:

- If MEP is disabled, the following algorithm methods default to Round Robin.
    - RTT
    - Least Connections
    - Least Bandwidth
    - Least Packets
    - Least Response Time
- In the static proximity GLSB method, the appliance sends the request to the IP address of the site that best matches the proximity criteria.
- In the Round Trip Time method, the dynamic round trip time (RTT) values are to select the IP address of the best performing site. RTT is a measure of the delay in the network between the client's local DNS server and a data resource.

GSLB Active-Active Datacenter Topology

In the diagram, Site 1 and Site 2 are active GLSB sites.


Site 1
DNS Traffic
ADNS IP
NetScaler GSLB
Public IP
NetScaler Load Balancer
Private IP
Internet
MEP
Sync
Site 2
ADNS IP
NetScaler GSLB
Public IP
NetScaler Load Balancer
Private IP
DNS Traffic

When the client sends a DNS request, it lands in one of the active sites.

If Site 1 receives the client request, the GSLB virtual server in Site 1 selects a load balancing or content switching virtual server and sends the virtual server's IP address to the DNS server, which sends it to the client. The client then resends the request to the new virtual server at the new IP address.

As both sites are active, the GSLB algorithm evaluates the services at both sites when making a selection as determined by the configured GSLB method.

Active-Passive Site Deployment

September 21, 2020

An active-passive site consists of an active and a passive data center. This deployment type is ideal for disaster recovery.

In this type of deployment, some of the sites (remote sites) are reserved only for disaster recovery. These sites do not participate in any decision making until all the active sites are DOWN. A passive site does not become operational unless a disaster event triggers a failover.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the passive GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

An active-passive deployment can include a maximum of 32 GSLB sites, because MEP cannot synchronize more than 32 sites.

For an active-passive deployment, you can configure the following GSLB methods.

- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

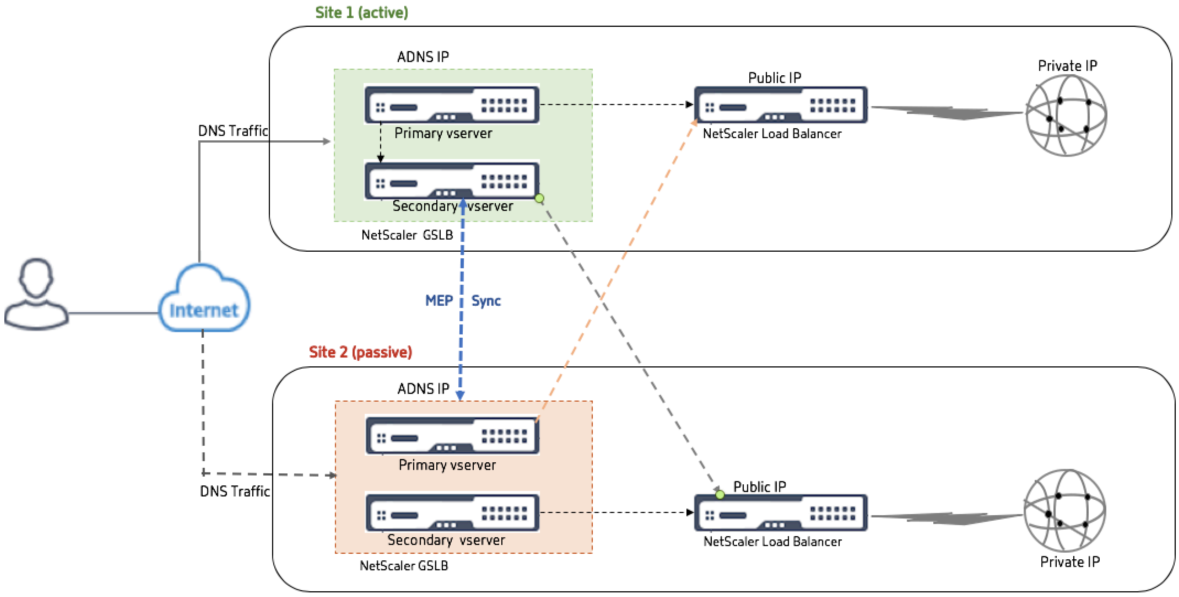
Note:

- If MEP is disabled, the following algorithm methods default to Round Robin.
  - RTT
  - Least Connections
  - Least Bandwidth
  - Least Packets
  - Least Response Time

- In the static proximity GLSB method, the appliance sends the request to the IP address of the site that best matches the proximity criteria.
- In the Round Trip Time method, the dynamic round trip time (RTT) values are to select the IP address of the best performing site. RTT is a measure of the delay in the network between the client's local DNS server and a data resource.

GSLB Active-Passive Datacenter Topology Diagram

In the diagram, Site 1 is an active site and Site 2 is a passive site, which has the same configuration as that of Site 1.


Site 1 (active)
ADNS IP
Public IP
Private IP
DNS Traffic
Primary vserver
NetScaler Load Balancer
Secondary vserver
NetScaler GSLB
Internet
MEP
Sync
Site 2 (passive)
ADNS IP
Primary vserver
Public IP
DNS Traffic
Secondary vserver
NetScaler Load Balancer
NetScaler GSLB
Private IP

If Site 1 goes DOWN, Site 2 becomes operational.

When the client sends a DNS request, the request can land in any of the sites. However, the services are selected only from the active site (Site1) as long as it is UP.

Services from the passive site (Site 2) are selected only if the active site (Site 1) is DOWN.

Parent-Child Topology Deployment using the MEP Protocol

November 2, 2020

NetScaler GSLB provides global server load balancing and disaster recovery by creating mesh connections between all the involved sites and making intelligent load balancing decisions. Each site communicates with the others to exchange server and network metrics through Metric Exchange Protocol (MEP), at regular intervals. However, with the increase in number of peer sites, the volume of

MEP traffic increases exponentially because of the mesh topology. To overcome this, you can use a parent-child topology. The parent-child topology also supports larger deployments. In addition to the 32 parent sites, you can configure 1024 child sites.

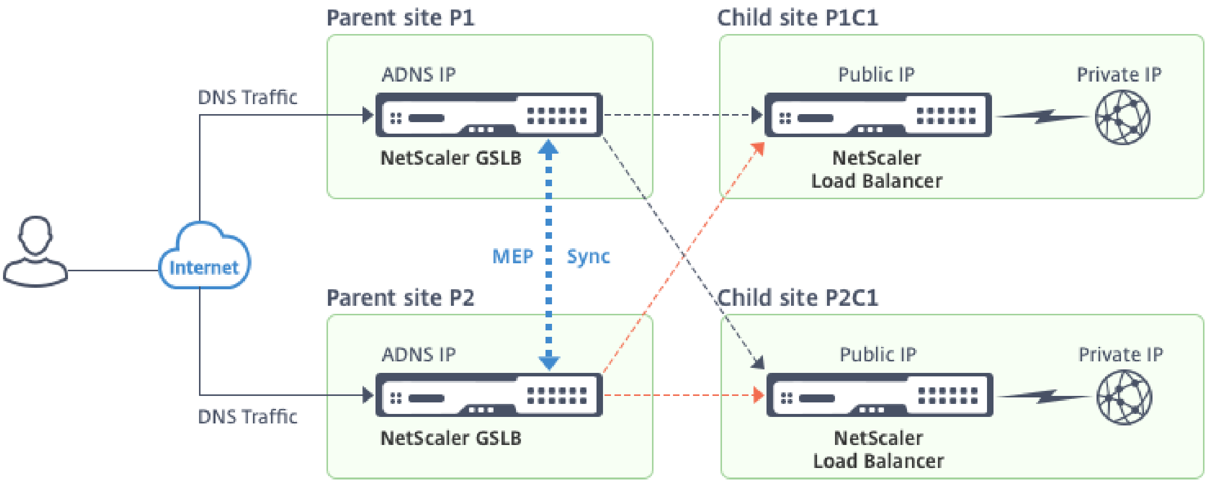
The GSLB parent-child topology is a two-level hierarchical design with the following characteristics:

- At the top level are parent sites, which have peer relationships with other parents.
- Each parent can have multiple child sites.
- Each parent site exchanges health information with its child sites and with other parent sites.
- A child site communicates only with its parent site.
- In a parent-child relationship for GSLB, only the parent site responds to ADNS queries. The child sites act as normal load balancing sites.
- An ADNS service or DNS load balancing virtual servers should be configured only in the parent site.
- A parent site can have a normal GSLB configuration, that is, services from local and all remote sites, but a child site can have local services only. Also, only the parent sites have GSLB virtual servers configured.

> **Note:**
>
> - In a parent-child topology, the exchange of site metrics is initiated from the lower of two IP addresses. However, from NetScaler release 11.1 build 51.x, the parent sites initiate connections to the child sites, and not vice versa, because the parent sites have information about all the child sites in the GSLB setup.
> - In a parent-parent connection, the exchange of site metrics is still initiated from the lower IP of two IP addresses.
> - In a parent-child topology, GSLB services are not always required to be configured on a child site. However, if you have additional configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the child site and configure it accordingly.
> - In a parent-child topology, the parent site and the child site can be on different NetScaler software versions. However, to use the GSLB automaticConfigSync option to synchronize the configuration across the parent sites, all parent sites must be on the same NetScaler software versions. If you are not using the automaticConfigSync option, then the parent site and the child site can be on different NetScaler software versions but make sure that you are not using any of the new features in the latest release. This is also applicable, in general, to two NetScaler nodes participating in GSLB.

Basic parent-child topology

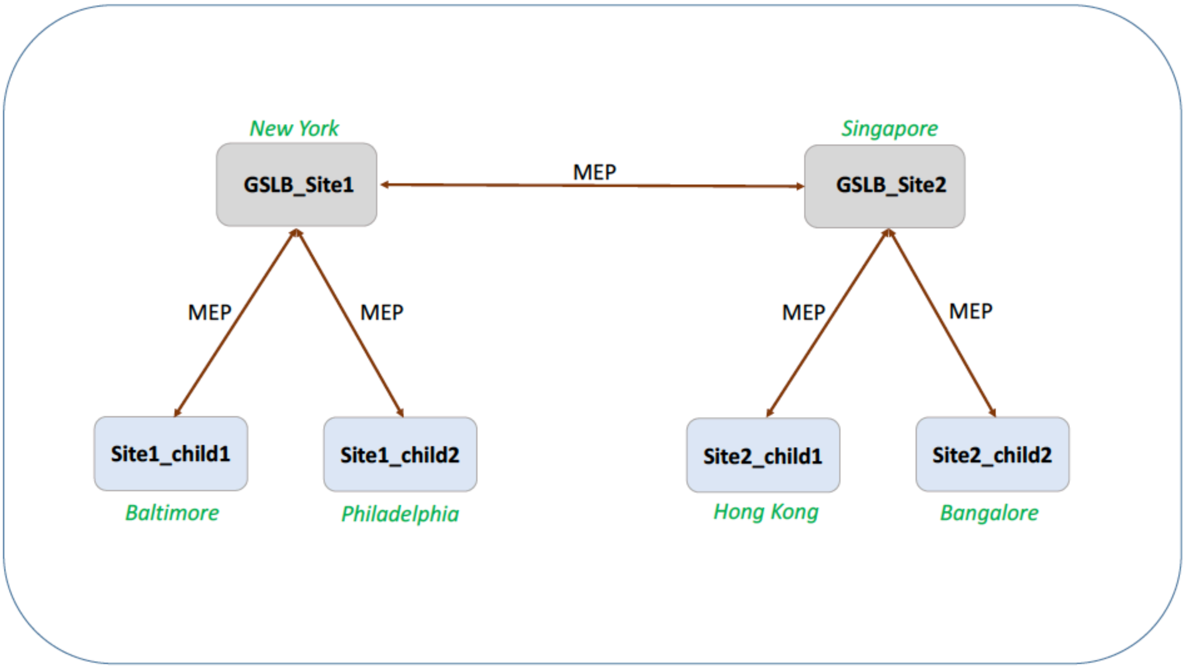
In the diagram, SiteP1 and SiteP2 are parent sites in a peer relationship. Site P1C1 and P2C1 are the child sites of P1 and P2 respectively.


Parent site P1
ADNS IP
NetScaler GSLB
Child site P1C1
Public IP
Private IP
NetScaler
Load Balancer
DNS Traffic
Internet
MEP
Sync
Parent site P2
ADNS IP
NetScaler GSLB
Child site P2C1
Public IP
Private IP
NetScaler
Load Balancer
DNS Traffic

Setting up a parent-child configuration for GSLB

If you have a firewall configured at a GSLB site, make sure that port 3011 is open.

The following diagram displays a sample parent-child configuration.


New York
GSLB_Site1
MEP
Singapore
GSLB_Site2
MEP
MEP
MEP
MEP
Site1_child1
Baltimore
Site1_child2
Philadelphia
Site2_child1
Hong Kong
Site2_child2
Bangalore

- The configuration of a child site includes the child site and its parent site, but no other parent or child sites.
- Network metrics, such as RTT and persistence session information, are synchronized only across the parent sites. Therefore, parameters such as nwMetricExchange and sessionExchange are disabled by default on all the child sites.
- To verify correct parent-child configuration, check the states of all the GSLB services bound to the parent sites.

To set up a parent-child configuration for GSLB by using the NetScaler command line

1. On each parent site, enter the following commands:

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>]

add gslb site<siteName> <siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

Example:

```
add gslb site GSLB_Site1 10.1.1.1 - publicIP 10.1.1.1

add gslb site Site1_child1 1 10.1.1.2 -publicIP 10.1.1.2 -parentSite GSLB_Site1
<!--NeedCopy-->
```

2. On each child site, enter the following commands:

```
add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>]

add gslb site <siteName> <siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>] [-parentSite <string>]
<!--NeedCopy-->
```

Example:

```
add gslb site* GSLB_Site1 10.1.1.1 - publicIP 10.1.1.1

add gslb site Site1_child1 1 10.1.1.2 -publicIP 10.1.1.2 -parentSite GSLB_Site1
<!--NeedCopy-->
```

For a complete example of a parent-child configuration, using the command line interface, see Example of a Complete Parent-Child Configuration, Using the NetScaler CLI.

> **Note**
>
> If the load balancing virtual server IP address is a private IP address and the public IP address is different from this IP address, you need to configure a GSLB service for the local load balancing virtual server on the child site. This is required for statistics collection between the parent and the child site.

> On the child site, at the command prompt, type:
>
> ```
> add gslb service <name> <private IP/lb vserver IP> http 80 -sitename ><
> childsite name> -publicip <public IP of LB vserver>
> ```
>
> **Example:**
>
> add gslb service Service-GSLB 192.168.1.3 http 80 -GSLB_Site11 site 11_lb1 172.16.1.1
>
> Where 192.168.1.3 is a private IP address of the load balancing virtual server and 172.16.1.1 is a public IP address of the load balancing virtual server.

Backing UP a Parent Site

Note: This feature was introduced in NetScaler release 11.1 build 51.x. To use the backup parent site topology, make sure that the parent site and the child sites are on NetScaler 11.1 build 51.x and later.

The backup parent site topology is useful in scenarios wherein a large number of child sites are associated with a parent site. If this parent site goes DOWN, all of its child sites become unavailable. To prevent this, you can now configure a backup parent site to which the child sites can connect if the original parent site is DOWN. The parent site sends the backup parent list to the child sites through the MEP messages.

When a parent site is DOWN, the other parent sites in the GSLB get to know that a particular parent site is DOWN through MEP because MEP to that parent site is DOWN. The other parent sites in the GSLB setup look up the backup chain of the peer parent. The parent site with the highest preference adopts the child sites of the parent that went DOWN. The new parent then initiates a connection with the child site. A child site can accept or reject the connection after evaluating its existing connections and the information in the backup list.
When the original parent site is back UP, it tries to establish connections with its child sites that have migrated to a different parent. When a connection attempt is successful, the child site is reassigned to its original parent site.

Note:

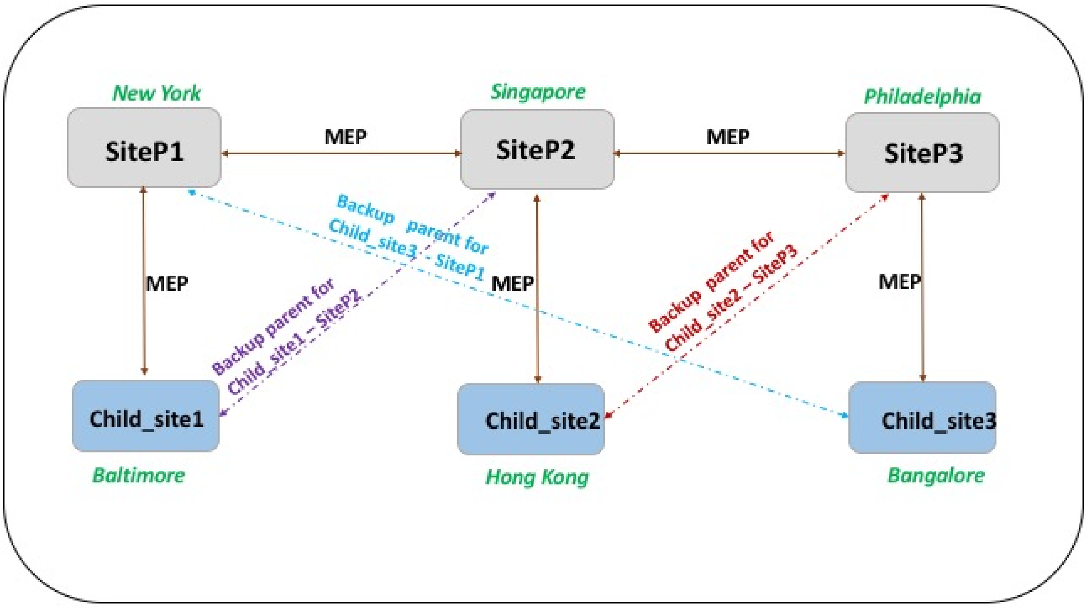
- Only parent sites can be configured as backups, and this configuration can only be done at the parent site.
- All child sites use the backup parent set.
- Synchronization is done only on the parent sites. GSLB child sites' configuration is not affected by synchronization. This is because the parent site and the child site configurations are not identical. The child sites configuration consists only of its own and its parent site's details. Also, GSLB services are not always required to be configured in the child sites.

Consider the configuration shown in the following figure, in which:

- SiteP1, SiteP2, and SiteP3 are the parent sites.

- Child_site1, Child_site2, and Child_site3 are the child sites of SiteP1, SiteP2, and SiteP3 respectively.
- Backup parent sites;
    - SiteP1 backup parents - SiteP2 (higher preference) and SiteP3
    - SiteP2 backup parents – SiteP3 (higher preference) and SiteP1
    - SiteP3 backup parents – SiteP1 (higher preference) and SiteP2

Note: For illustration purposes, the figure shows only one backup parent for each parent site.


New York
SiteP1
MEP
Singapore
SiteP2
MEP
Philadelphia
SiteP3
MEP
Backup parent for Child_site3 - SiteP1
MEP
Backup parent for Child_site1 – SiteP2
Backup parent for Child_site2 – SiteP3
MEP
Child_site1
Child_site2
Child_site3
Baltimore
Hong Kong
Bangalore

The following list summarizes the behavior of the parent and child sites under various scenarios:

- Scenario 1: SiteP1 goes DOWN.
    - SiteP2 and SiteP3 detect that SiteP1's MEP connection is DOWN. SiteP2 is higher in the preference list of backup parents for SiteP1, so it tries to initiate a connection to Child_site1. SiteP3 assumes that Child_site1 is now the child site of parent SiteP2.
    - SiteP2 sends Child_Site1 the list of SiteP1's backup parents (SiteP2 and SiteP3) to Child_site1. Child_site1 uses the list to decide whether to accept or reject the connection from SiteP2. It accepts the connection and becomes a child of SiteP2.
    - When SiteP1 is back UP, it sends Child_site1 a connection request. The new request takes precedence and Child_site 1 migrates to SiteP1.

- Scenario 2: Only the MEP connection between SiteP1 and SiteP2 has gone DOWN. Child_site1 rejects SiteP2's connection request, because its parent, SiteP1, is still UP.

- Scenario 3: SiteP3 and Child_Site1 detect that SiteP1 is DOWN, and the MEP connection between SiteP3 and SiteP2 is also DOWN. However, SiteP2 detects that SiteP1 is UP, and the MEP connection between SiteP1 and SiteP2 is UP.

  - SiteP2 does not take any action.
  - SiteP3 checks SiteP1's backup list and finds that SiteP2 has a higher preference than SiteP3. But SiteP2 is DOWN, so SiteP3 tries to establish a connection with Child_site1. Child_site1 has detected that SiteP1 is DOWN, so it accepts SiteP3's connection request.
  - Now the connection between SiteP1 and SiteP2 goes DOWN. SiteP2 checks SiteP1's backup list and finds itself as the most preferred backup, so it tries to connect to Child_site1. Child_site1 evaluates the new connection request based on SiteP1's list and finds SiteP2 as the most preferred backup, so it migrates to SiteP2 from SiteP3.

To configure a backup parent site by using the command line interface

At the command prompt type:

```
set gslb site <sitename> -backupParentlist <bkp_site1> <bkp_site2> … <bkp_site5>
<!--NeedCopy-->
```

Example:

For the parent site (SiteP1), the sites (SiteP2 and SiteP3) are configured as the backup parent sites.

```
set gslb site SiteP1 -backupParentlist SiteP2 SiteP3
<!--NeedCopy-->
```

Note:

- You cannot add a new site as a backup parent. You must first add all the sites, and then configure the site as a backup parent.
- To remove a backup parent, you must use the unset command, which unsets all the sites that were previously configured as backup parent sites.

To configure a backup parent site by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Sites**.
2. Add a new site or select an existing site.
3. Choose the **Backup Parent Sites** option box while creating or configuring the GSLB site.

GSLB Configuration Entities

September 21, 2020

A GSLB configuration consists of a group of GSLB entities on each appliance in the configuration. These entities include the following:

- GSLB Sites
- GSLB Services
- GSLB Virtual Servers
- Load Balancing or Content Switching Virtual Servers
- ADNS Services
- DNS VIPs

GSLB Sites

A typical GSLB setup consists of data centers, each of which has various network appliances that may or may not be NetScaler appliances. The data centers are called GSLB sites. Each GSLB site is managed by a NetScaler appliance that is local to that site. Each of these appliances treats its own site as the local site and all other sites, managed by other appliances, as remote sites.

If the appliance that manages a site is the only NetScaler appliance in that data center, the GSLB site hosted on that appliance acts as a bookkeeping placeholder for auditing purposes, because no metrics can be collected. Typically, this happens when the appliance is used only for GSLB, and other products in the data center are used for load balancing or content switching.

Relationships among GSLB Sites

The concept of sites is central to NetScaler GSLB implementations. Unless otherwise specified, sites form a peer relationship among themselves. This relationship is used first to exchange health information and then to distribute load as determined by the selected algorithm. In many situations, however, a peer relationship among all GSLB sites is not desirable. Reasons for not having an all-peer implementation could be;

- To clearly separate GSLB sites. For example, to separate sites that participate in resolving DNS queries from the traffic management sites.
- To reduce the volume of Metric Exchange Protocol (MEP) traffic, which increases exponentially with an increasing number of peer sites.

These goals can be achieved by using parent and child GSLB sites.

GSLB Services

A GSLB service is usually a representation of a load balancing or content switching virtual server, although it can represent any type of virtual server. The GSLB service identifies the virtual server's IP address, port number, and service type. GSLB services are bound to GSLB virtual servers on the

NetScaler appliances managing the GSLB sites. A GSLB service bound to a GSLB virtual server in the same data center is local to the GSLB virtual server. A GSLB service bound to a GSLB virtual server in a different data center is remote from that GSLB virtual server.

> **Note**
>
> Sites and services are inherently linked to indicate proximity between the two. That is, all services must belong to a site, and are assumed to be in the same location as the GSLB site for proximity purposes. Likewise, services and virtual servers are linked, so that the logic is linked to the resources that are available.

GSLB Virtual Servers

A GSLB virtual server has one or more GSLB services bound to it, and load balances traffic among those services. It evaluates the configured GSLB methods (algorithms) to select the appropriate service to which to send a client request. Because the GSLB services can represent either local or remote servers, selecting the optimal GSLB service for a request has the effect of selecting the data center that should serve the client request.

The domain for which global server load balancing is configured must be bound to the GSLB virtual server, because one or more services bound to the virtual server will serve requests made for that domain.

Unlike other virtual servers configured on a NetScaler appliance, a GSLB virtual server does not have its own virtual IP address (VIP).

Load Balancing or Content Switching Virtual Servers

A load balancing or content switching virtual server represents one or many physical servers on the local network. Clients send their requests to the load balancing or content switching virtual server's virtual IP (VIP) address, and the virtual server balances the load across the physical servers. After a GSLB virtual server selects a GSLB service representing either a local or a remote load balancing or content switching virtual server, the client sends the request to that virtual server's VIP address.

For more information about load balancing or content switching virtual servers and services, see Load Balancing or Content Switching.

ADNS Services

An ADNS service is a special kind of service that responds only to DNS requests for domains for which the NetScaler appliance is authoritative. When an ADNS service is configured, the appliance owns that IP address and advertises it. Upon reception of a DNS request by an ADNS service, the appliance checks for a GSLB virtual server bound to that domain. If a GSLB virtual server is bound to the domain, it is queried for the best IP address to which to send the DNS response.

DNS VIPs

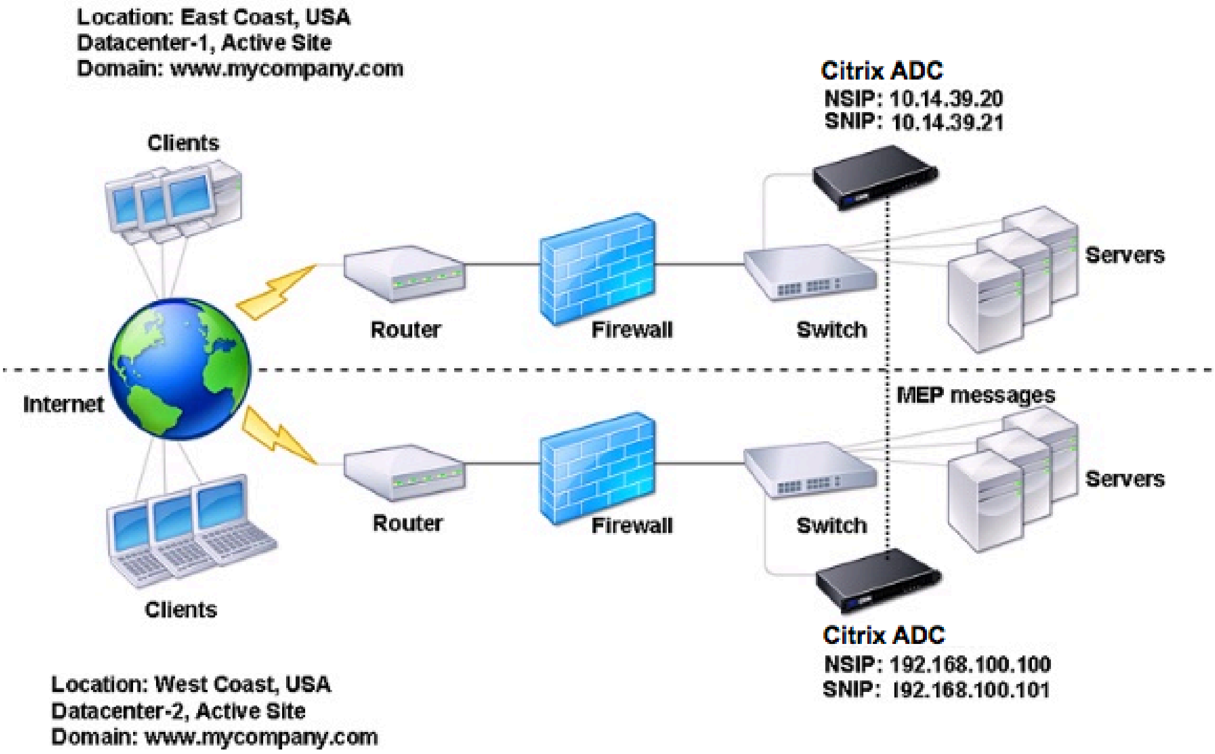
A DNS virtual IP is a virtual IP (VIP) address that represents a load balancing DNS virtual server on the NetScaler appliance. DNS requests for domains for which the NetScaler appliance is authoritative can be sent to a DNS VIP.

Configuring Global Server Load Balancing (GSLB)

September 21, 2020

Global server load balancing is used to manage traffic flow to a web site hosted on two separate server farms that ideally are in different geographic locations. For example, consider a Web site, www.mycompany.com, which is hosted on two geographically separated server farms or data centers. Both server farms use NetScaler appliances. The NetScaler appliances in these server farms are set up in one-arm mode and function as authoritative DNS servers for the www.mycompany.com domain. The following figure illustrates this configuration.

Figure 1. Basic GSLB Topology


Location: East Coast, USA
Datacenter-1, Active Site
Domain: www.mycompany.com
Citrix ADC
NSIP: 10.14.39.20
SNIP: 10.14.39.21
Clients
Servers
Router
Firewall
Switch
Internet
MEP messages
Servers
Router
Firewall
Switch
Clients
Citrix ADC
NSIP: 192.168.100.100
SNIP: 192.168.100.101
Location: West Coast, USA
Datacenter-2, Active Site
Domain: www.mycompany.com

To configure such a GSLB setup, you must first configure a standard load balancing setup for each server farm or data center. This enables you to balance load across the different servers in each server farm. Then, configure both NetScaler appliances as authoritative DNS (ADNS) servers. Next, create a GSLB site for each server farm, configure GSLB virtual servers for each site, create GLSB services, and bind the GSLB services to the GSLB virtual servers. Finally, bind the domain to the GSLB virtual servers. The GSLB configurations on the two appliances at the two different sites are identical, although the load-balancing configurations for each site is specific to that site.

Note: To configure a GSLB site in a NetScaler cluster setup, see
Setting Up GSLB in a Cluster.

Configuring a Standard Load Balancing Setup

A load balancing virtual server balances the load across different physical servers in the data center. These servers are represented as services on the NetScaler appliance, and the services are bound to the load balancing virtual server.

For details on configuring a basic load balancing setup, see Load Balancing.

Configuring an Authoritative DNS Service

September 21, 2020

When you configure the NetScaler appliance as an authoritative DNS server, it accepts DNS requests from the client and responds with the IP address of the data center to which the client should send requests.

Note: For the NetScaler appliance to be authoritative, you must also create SOA and NS records. For more information about SOA and NS records, see
Domain Name System.

To create an ADNS service by using the command line interface

At the command prompt, type the following commands to create an ADNS service and verify the configuration:

```
add service <name> <IP>@ ADNS <port>

show service <name>
<!--NeedCopy-->
```

Example:

```
add service Service-ADNS-1 10.14.39.21 ADNS 53
```

```

show service Service-ADNS-1
<!--NeedCopy-->
```

To modify an ADNS service by using the command line interface

At the command prompt, type the following command:

```
set service <name> <IPAddress> ADNS <port>
<!--NeedCopy-->
```

Example:

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
<!--NeedCopy-->
```

To remove an ADNS service by using the command line interface

At the command prompt, type the following command:

```
rm service <name>
<!--NeedCopy-->
```

Example:

```
rm service Service-ADNS-1
<!--NeedCopy-->
```

To configure an ADNS service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Add a new ADNS service, or select an existing service and edit its settings.

Configuring a Basic GSLB Site

September 21, 2020

A GSLB site is a representation of a data center in your network and is a logical grouping of GSLB virtual servers, services, and other network entities. Typically, in a GSLB set up, there are many GSLB sites that are equipped to serve the same content to a client. These are usually geographically separated to ensure that the domain is active even if one site goes down completely. All of the sites in the GSLB configuration must be configured on every NetScaler appliance hosting a GSLB site. In other words, at each site, you configure the local GSLB site and each remote GSLB site.

Once GSLB sites are created for a domain, the NetScaler appliance sends client requests to the appropriate GSLB site as determined by the GSLB algorithms configured.

To create a GSLB site by using the command line interface

At the command prompt, type the following commands to create a GSLB site and verify the configuration:

```
add gslb site <siteName> <siteIPAddress>
show gslb site <siteName>
<!--NeedCopy-->
```

Example:

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show glsb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To modify or remove a GSLB Site by using the command line interface

- To modify a GSLB site, use the set gslb site command, which is just like using the add gslb site command, except that you enter the name of an existing GSLB Site.
- To unset a site parameter, use the unset gslb site command, followed by the siteName value and the name of the parameter to be reset to its default value.
- To remove a GSLB site, use the rm gslb site command, which accepts only the `<name>` argument.

To configure a basic GSLB site by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. Add a new GSLB site, or select an existing GSLB site and edit its settings.

To view the statistics of a GSLB site by using the command line interface

At the command prompt, type:

```
stat gslb site <siteName>
<!--NeedCopy-->
```

Example:

```
stat gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To view the statistics of a GSLB site by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. Select the GSLB site and click **Statistics**.

Configuring a GSLB Service

September 21, 2020

A GSLB service is a representation of a load balancing or content switching virtual server. A local GSLB service represents a local load balancing or content switching virtual server. A remote GSLB service represents a load balancing or content switching virtual server configured at one of the other sites in the GSLB setup. At each site in the GSLB setup, you can create one local GSLB service and any number of remote GSLB services.

> **Important**
>
> If the load balancing virtual server is either in a GSLB node itself or is in a child node (in parent-child deployment) and no monitors are bound to the GSLB service, then make sure the following:
>
> The GLSB service IP address, port number, and protocol match the virtual server that the service is representing. Else, the service state is marked as DOWN.

To create a GSLB service by using the command line interface

At the command prompt, type the following commands to create a GSLB service and verify the configuration:

```
add gslb service <serviceName> <serverName | IP> <serviceType> <port>-siteName <string>
show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80  - siteName Site-GSLB-East-Coast
```

```
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

To modify or remove a GSLB service by using the command line interface

- To modify a GSLB service, use the set gslb service `<serviceName>` command. For this command, specify the name of the GSLB service whose configuration you want to modify. You can change the existing values of the parameters either specified by you or set by default. You can change the value of more than one parameter in the same command. Refer to the add gslb service command for details about the parameters. Example

  ```
  > set gslb service SKP_GSLB_NOTCNAME_SVC2 -maxBandWidth 25 -
     maxClient 8
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 25 kbits
  <!--NeedCopy-->
  ```

- To reset a parameter to its default value, you can use the unset gslb service `<serviceName>` command and the parameters to be unset. Example

  ```
  > unset gslb service SKP_GSLB_NOTCNAME_SVC2 maxBandWidth
  Done
  > sh gslb service SKP_GSLB_NOTCNAME_SVC2
  SKP_GSLB_NOTCNAME_SVC2 (21.211.21.21: 80)- HTTP
  ...
  Max Conn: 8  Max Bandwidth: 0 kbits
  <!--NeedCopy-->
  ```

- To remove a GSLB service, use the rm gslb service `<serviceName>` command.

To create a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Add a new GSLB service, or select an existing service and edit its settings.

To view the statistics of a GSLB service by using the command line interface

At the command prompt, type:

```
stat gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
stat gslb service Service-GSLB-1
<!--NeedCopy-->
```

To view the statistics of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Select the GSLB Service and click **Statistics**.

Configuring a GSLB Virtual Server

September 21, 2020

A GSLB virtual server is an entity that represents one or more GSLB services and balances traffic between them. It evaluates the configured GSLB methods or algorithms to select a GSLB service to which to send the client request.

> **Note**
>
> A GSLB virtual server protocol requirement is mainly to create a relation between the virtual server and the services that are bound to the virtual server. This also keeps CLI/APIs consistent for other types of virtual servers. The Service Type parameter on a service or a virtual server is not used while serving the DNS requests. It is instead referenced during site persistence, monitoring, and for doing lookups via MEP.

To create a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to add a GSLB virtual server and verify the configuration:

```
-  add gslb vserver <name> <serviceType> -ipType (IPv4 | IPv6)
-  show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
add gslb vserver Vserver-GSLB-1 HTTP -ipType IPv4
add gslb vserver Vserver-GSLB-2 HTTP -ipType IPv6
show gslb vserver Vserver-GSLB-1
show gslb vserver Vserver-GSLB-2
<!--NeedCopy-->
```

To modify or remove a GSLB virtual server by using the command line interface

- To modify a GSLB virtual server, use the `set gslb vserver` command. This command works similar to the `add gslb vserver` command, except that you enter the name of an existing GSLB virtual server.
- To reset a parameter to is default value, you can use the `unset gslb vserver` command followed by the vserverName value and the name of the parameter to be unset.
- To remove a GSLB virtual server, use the `rm gslb vserver` command, which accepts only the name argument.

To configure a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Add a new GSLB virtual server, or select an existing GSLB virtual server and edit its settings.

To view the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

```
stat gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
stat gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To view the statistics of a GSLB virtual server by using the configuration utility

Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**, select the virtual server and click **Statistics**.

Statistics of a GSLB service

When you run the stat gslb service command from the command line or click the **Statistics link** from the configuration utility, the following details of the service are displayed:

- **Request bytes.** Total number of request bytes received on this service or virtual server.
- **Response bytes.** Number of response bytes received by this service or virtual server.
- **Current client established connections.** Number of client connections in ESTABLISHED state.
- **Current load on the service.** Load on the service (Calculated from the load monitor bound to the service).

The data of number of requests and responses, and the number of current client and server connections may not be displayed or may not be synchronized with the data of the corresponding load balancing virtual server.

Clearing the GSLB virtual server or service statistics

You can now clear the statistics of a GSLB virtual server and service. NetScaler provides the following two options to clear the statistics:

- **Basic**: Clears the statistics that are specific to the virtual server but retains the statistics that are contributed by the bound GLSB service.
- **Full**: Clears both the virtual server and the bound GSLB service statistics.

To clear the statistics of a GSLB virtual server by using the command line interface

At the command prompt, type:

```
stat gslb vserver <name> -clearstats <basic | full>
<!--NeedCopy-->
```

Example:

```
stat gslb vserver Vserver-GSLB-1  - clearstats basic
<!--NeedCopy-->
```

To clear the statistics of a GSLB service by using the command line interface

At the command prompt, type:

```
stat gslb service <name> -clearstats <basic | full>
<!--NeedCopy-->
```

Example:

```
stat gslb service service-GSLB-1  -clearstats basic
<!--NeedCopy-->
```

To clear the statistics of a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Select the GSLB virtual server and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

To clear the statistics of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Select the GSLB service and, click **Statistics**, and then click **Clear**.
3. From the **Clear** drop-down list, select **Basic** or **Full**, and then click **OK**.

Enabling and Disabling GSLB Virtual Servers

When you create a GSLB virtual server, it is enabled by default. If you disable the GSLB virtual server, upon receiving a DNS request, the NetScaler appliance does not make any GSLB decision based on the GSLB method that is configured. Instead, the response to the DNS query contains the IP addresses of all the services bound to the virtual server.

To enable or disable a GSLB virtual server by using the command line interface

At the command prompt, type one of the following commands:

```
enable gslb vserver <name>@

disable gslb vserver <name>@
<!--NeedCopy-->
```

Example:

```
enable gslb vserver Vserver-GSLB-1
disable gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To enable or disable a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. Select a virtual server and, from the **Action** list, select **enable** or **disable**.

Binding GSLB Services to a GSLB Virtual Server

September 21, 2020

Once the GSLB services and virtual server are configured, relevant GSLB services must be bound to the GSLB virtual server to activate the configuration.

To bind a GSLB service to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a GSLB service to a GSLB virtual server and verify the configuration:

```
bind gslb vserver <name> -serviceName <string>

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To unbind a GSLB service from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -serviceName <string>
<!--NeedCopy-->
```

To bind GSLB services by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click a virtual server.
2. Click in the **Domains** section, and configure a domain and bind the domain.

Binding a Domain to a GSLB Virtual Server

September 21, 2020

To make a NetScaler appliance the authoritative DNS server for a domain, you must bind the domain to the GSLB virtual server. When you bind a domain to a GSLB virtual server, the NetScaler appliance adds an address record for the domain, containing the name of the GSLB virtual server. The start of authority (SOA) and name server (NS) records for the GSLB domain must be added manually.

For details on configuring SOA and NS records, see Domain Name System.

To bind a domain to a GSLB virtual server by using the command line interface

At the command prompt, type the following commands to bind a domain to a GSLB virtual server and verify the configuration:

```
bind gslb vserver <name> -domainName <string>
show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To unbind a GSLB domain from a GSLB virtual server by using the command line interface

At the command prompt, type:

```
unbind gslb vserver <name> -domainName <string>
<!--NeedCopy-->
```

To bind a domain to a GSLB virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server to which you want to bind the domain (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, do one of the following:
    - To create a new Domain, click **Add**.
    - To modify an existing Domain, select the domain, and then click **Open**.
4. In the Create GSLB Domain or Configure GSLB Domain dialog box, specify values for the following parameters as shown:

   - Domain Name*—domainName (for example, www.mycompany.com)

   /* A required parameter

5. Click Create.
6. Click OK.

To view the statistics of a domain by using the command line interface

At the command prompt, type:

```
stat gslb domain <name>
<!--NeedCopy-->
```

Example:

```
stat gslb domain www.mycompany.com
<!--NeedCopy-->
```

Note: To view statistics for a particular GSLB domain, enter the name of the domain exactly as it was added to the NetScaler appliance. If you do not specify the domain name, or if you specify an incorrect domain name, statistics for all configured GSLB domains are displayed.

To view the statistics of a domain by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In GSLB Virtual Servers pane, select the GSLB Virtual Server (for example, Vserver-GSLB-1) and click Open.
3. In the Configure GSLB Virtual Server dialog box, on the Domains tab, select the domain, and then click **Statistic**s.

To view the configuration details of the entities bound to a GSLB domain using the command line

Note: This feature is available in NetScaler release 10.5.e.

At the command prompt, type:

```
show gslb domain <name>
<!--NeedCopy-->
```

Example:

```
show gslb domain gslb1.com
       gslb1.com
       gvs1 - HTTP    state: DOWN
       DNS Record Type: A
       Configured Method: LEASTCONNECTION
       Backup Method: ROUNDROBIN
       Persistence Type: NONE
       Empty Down Response: DISABLED
       Multi IP Response: DISABLED
       Dynamic Weights: DISABLED

       gsvc1 (10.102.239.165: 80)- HTTP State: DOWN    Weight: 1
              Dynamic Weight: 0    Cumulative Weight: 1
              Effective State: DOWN
              Threshold : BELOW

              Monitor Name : http
                     State: DOWN   Weight: 1
                     Probes: 144   Failed [Total: 144 Current: 144]
                     Last response: Failure - TCP syn sent, reset received.
                     Response Time: 2000 millisec

       gsvc2 (10.102.239.179: 80)- HTTP State: DOWN    Weight: 1
              Dynamic Weight: 0    Cumulative Weight: 1
              Effective State: DOWN
              Threshold : BELOW

              Monitor Name : http-ecv
                     State: DOWN   Weight: 1
                     Probes: 141   Failed [Total: 141 Current: 141]
                     Last response: Failure - TCP syn sent, reset received.
                     Response Time: 2000 millisec
Done
<!--NeedCopy-->
```

To view the configuration details of the entities bound to a GSLB domain by using the configuration utility

Note: This feature is available in NetScaler release 10.5.e.

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click a virtual server.

2. Click the field below the **Domains** pane.
3. In the **GSLB Virtual Server Domain Binding** dialog box, select a domain, and then click **Show Bindings**.

Synchronizing a Configuration in a GSLB Setup

September 21, 2020

Typically, a GSLB setup has a few data centers with a GSLB site configured for each data center. In each NetScaler, participating in GSLB, configure one GSLB site as a local site and the others as remote sites. When you add another GSLB site at a later point of time, ensure that all the GSLB sites have the same configuration. To have the same configuration on all the GSLB sites, you can use the NetScaler appliance's GSLB configuration synchronization option.

The NetScaler appliance from which you use the synchronization option is referred to as the 'master node' and the GSLB sites on which the configuration is copied as 'slave nodes'. When you synchronize a GSLB configuration, the configurations on all the GSLB sites participating in the GSLB setup are made similar to that on the master node.

Synchronization (may also be referred to as 'auto sync') is carried out in the following manner:

- The master node finds the differences between the configuration of the master node and slave node, and changes the configuration of the slave node to make it similar to the master node. If you force a synchronization (use the 'force sync' option), the NetScaler deletes the GSLB configuration from the slave node and then configures the slave to make it similar to the master node.
- During synchronization, if a command fails, synchronization is not aborted and the error message are logged into a **.err** file in the **/var/netscaler/gslb** directory.
- Synchronization is done only on the parent sites. GSLB child sites' configuration is not affected by synchronization. This is because the parent site and the child site configurations are not identical. The child sites configuration consists only of its own and its parent site's details. Also, GSLB services are not always required to be configured in the child sites.
- If you disable the internal user login, the GSLB auto sync uses the SSH keys to synchronize the configuration. But, to use GSLB auto sync in partition environment, you need to enable the internal user login and make sure that the partition username in the local and remote GSLB sites is same.
- For enhanced security, Citrix recommends that you change the internal user account and RPC node passwords. Internal user account password is changed through RPC node password. For details, see Change an RPC node password.

Note:

- On the remote GSLB site RPC node, configure the firewall to accept auto-sync connections by specifying the remote site IP (cluster IP address for cluster setup) and port (3010 for RPC and 3008 for secure RPC). The source IP address that will be used for auto-sync is the NSIP of the master node (NSIP of the configuration coordinator in a cluster setup). The destination IP is the site IP (remote site IP).
- The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for a RPC node is specific to each NetScaler appliance. Therefore, after you force a synchronization (using the sync gslb config -forceSync command or by selecting the ForceSync option in the NetScaler GUI), you have to manually change the source IP addressess on the other NetScaler appliances.
  Port 22 is also required for synchronizing the database files to the remote site.

If you use the saveconfig option, the sites that participate in the synchronization process automatically save their configuration, in the following way:

1. The master node saves its configuration immediately before it initiates the process of synchronization.
2. After the process of synchronization is complete, the slave nodes save their configuration. A slave node saves its configuration only if the configuration difference was applied successfully on it. If synchronization fails on a slave node, you must manually investigate the cause of the failure and take corrective action.

Limitations of synchronization

- On the master node, the names of the remote GSLB sites must be identical to the names of sites configured on the NetScaler appliances hosting those sites.
- During the synchronization, traffic disruptions may occur.
- NetScaler can synchronize only up to 80000 lines of the configuration.
- Synchronization may fail:
  - If the spill over method is changed from CONNECTION to DYNAMIC CONNECTION.
  - If you interchange the site prefix of the GSLB services bound to a GSLB virtual server on the master node and then try to synchronize.
  - If the RPC node passwords are different for NetScaler IP address (NSIP) and loopback IP address.
  - If you perform synchronization on GSLB sites that are configured in different partitions of the same NetScaler appliance.
- If you have configured the GSLB sites as High Availability (HA) pairs, the RPC node passwords of primary and secondary nodes should be same.
- If you rename any GLSB entity that are part of your GSLB configuration (use “show gslb runningConfig” command to display the GSLB configuration). You need to use the force sync option to synchronize the configuration to other GSLB sites.

Note: To overcome the limitations due to some settings in the GSLB configuration, you can use the force sync option. But, if you use the force sync option the GSLB entities are removed and re-added to the configuration and the GSLB statistics are reset to zero. Hence the traffic is disrupted during the configuration change.

Points to note before starting the synchronization of a GSLB setup

Before you start the synchronization of a GSLB setup, make sure that:

- On all the GSLB sites including the master node, management access and SSH should be enabled for the IP address of the corresponding GSLB site. The IP address of a GSLB site must be an IP address owned by the NetScaler.

  For more information about adding the GSLB site IP addresses and enabling Management Access, see Configuring a Basic GSLB Site.

- The GSLB configuration on the NetScaler appliance that is considered as the master node is complete and appropriate to be copied on all the sites.
- If you are synchronizing the GSLB configuration for the first time, all the sites participating in GSLB need to have the GSLB site entity of their respective local sites.
- You are not synchronizing sites that, by design, do not have the same configuration.

> **Important**
>
> After a GSLB configuration is synchronized, the configuration cannot be rolled back on any of the GSLB sites. Run the sync gslb config command only if you are sure that the synchronization process will not overwrite the configuration on the remote site. Site synchronization is undesirable when the local and remote sites have different configurations by design, and can lead to site outage. If some commands fail and some commands succeed, the successful commands cannot be rolled back.

To synchronize a GSLB configuration by using the command line interface

At the command prompt, type the following commands to synchronize GSLB sites and verify the configuration:

```
sync gslb config [-preview | -forceSync <string> | -nowarn | -
    saveconfig] [-debug]

show gslb syncStatus
<!--NeedCopy-->
```

Example

```
> sync gslb config
[WARNING]: Syncing config may cause configuration loss on other site.
Please confirm whether you want to sync-config (Y/N)? [N]:y
Sync Time: Dec 9 2011 10:56:9
Retrieving local site info: ok
Retrieving all participating gslb sites info: ok
Gslb_site1[Master]:
        Getting Config: ok
Gslb_site2[Slave]:
        Getting Config: ok
        Comparing config: ok
         Applying changes: ok
Done
<!--NeedCopy-->
```

To synchronize a GSLB configuration by using the NetScaler GUI

Navigate to **Traffic Management** > **GSLB** and, under **GSLB Configuration**, click **Synchronize configuration on remote sites** and synchronize the GSLB configuration.

Previewing GSLB synchronization

Note: This feature was introduced in NetScaler release 11.1 build 42.5.

By previewing the GSLB synchronization operation, you can see the differences between the master node and each slave node. If there are any discrepancies, you can troubleshoot before synchronizing the GSLB configuration.

To preview the GSLB synchronization output by using the command line interface

At the command prompt, type the following command:

```
sync gslb config -preview
<!--NeedCopy-->
```

To preview the GSLB synchronization output by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **GSLB Configuration** > **Synchronize configuration on remote sites**.
2. Select the **Preview** check box.

3. Click **Run**.
   A progress window displays any discrepancies in the configuration.

Debugging the commands triggered during synchronization process

You can view the status (success or failure) of each command triggered during the synchronization process and troubleshoot accordingly.

To debug the GSLB synchronization commands by using the command line interface

At the command prompt, type the following command:

```
sync gslb config -debug
<!--NeedCopy-->
```

To debug the GSLB synchronization commands by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **GSLB Configuration** > **Synchronize configuration on remote sites**.
2. Select the **Debug** check box.
3. Click **Run**.
   A progress window displays the status of each command triggered during synchronization.

Real-time synchronization between sites participating in GSLB

Note: This feature was introduced in NetScaler release 11.1 build 51.x.

If you want to synchronize GSLB configuration across slave sites automatically when the commands are executed on master sites, you can now use the AutomaticConfigSync option to automatically synchronize the real-time GSLB configuration. You do not have to manually trigger the AutoSync option to synchornize the configuration.

If you attempt to manually synchronize (with the sync gslb config command) a site while it is being autosynchronized, a "Sync in progress" error message appears. Autosynchronization cannot be triggered for a site that is in the process of being synchronized manually.

Notes:

- All logs related to real-time sync are stored in the /var/netscaler/gslb/periodic_sync.log file.
- The sync status file and default configuration file are stored in the location /var/netscaler/gslb_sync.
- Enabling **AutomaticConfigSync** from default partition of a partitioned appliance is not supported. However, it can be enabled from all other partitions. The sync status file and default configuration file are stored in the location /var/partitions/<partition name>/netscaler/gslb_sync.

To enable real-time synchronization by using the command line interface

At the command prompt, type the following command:

```
set gslb parameter –automaticConfigSync          DISABLED)
(ENABLED
```

To enable real-time synchronization by using the NetScaler GUI

1. Navigate to **Configuration** >**Traffic Management** > **GSLB** > **Change GSLB Settings**.
2. Select **Automatic Config Sync** check box.

   Note: This option must be enabled only in the site where the configuration is performed.

Best practices for using the real-time synchronization feature

- It is recommended that all the NetScaler appliances participating as sites have the same NetScaler software version.
- To change the RPC node password, first change the password on the slave site and then on the master site.
- Configure local GSLB sites on each site participating in GSLB.
- Enable automaticConfigSync on one of the sites where the configuration is performed. This site eventually gets synchronized to other GSLB sites.
- If there is a new configuration or changes are made to the existing configuration, make sure to check the status using the "show gslb syncStatus" command to confirm if the changes are synchronized across all sites or if there was any error.

Testing the GSLB setup

September 21, 2020

Note: This feature was introduced in NetScaler release 11.1 build 51.x. This is a GUI only feature.

You can test the GSLB setup to make sure that the ADNS services or the DNS servers are responding with the correct IP address for the domain name that is configured in the GSLB setup.

To test the GSLB setup by using the NetScaler GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB**.
2. Click **Test GSLB**.

3. In the **Domain Name** box, select the domain name that you want to test.
4. In the **ADNS Service** or the **DNS Server** box, select one of the services or servers in the domain, and select the DNS record type, and then click **Test**.

A progress window appears in which the commands are triggered and the domain name that you selected for testing and its corresponding IP address based on GSLB configuration is displayed. You can check your configuration and make sure that the IP address is valid.

Repeat step 4 to test each of the services or servers.

Configuring the Metrics Exchange Protocol (MEP)

November 2, 2020

The data centers in a GSLB setup exchange metrics with each other through the metrics exchange protocol (MEP), which is a proprietary protocol for Citrix NetScaler appliance. The exchange of the metric information begins when you create a GSLB site. These metrics comprise load, network, and persistence information.

MEP is required for health checking of data centers to ensure their availability. A connection for exchanging network metric (round-trip time) can be initiated by either of the data centers involved in the exchange, but a connection for exchanging site metrics is always initiated by the data center with the lower IP address. By default, the data center uses a subnet IP address (SNIP) to establish a connection to the IP address of a different data center. However, you can configure a specific SNIP or virtual IP (VIP) address, or the NetScaler IP (NSIP) address, as the source IP address for metrics exchange. The communication process between GSLB sites uses TCP port 3011 or 3009, so this port must be open on firewalls that are between the NetScaler appliances.

Note: You can configure a SNIP or a GSLB site IP address as the source IP address for metrics exchange. For more information, see Configure source IP address for an RPC node.

If the source and target sites (the site that initiates a MEP connection and the site that receives the connection request, respectively) have both private and public IP addresses configured, the sites exchange MEP information by using the public IP addresses.

You can also bind monitors to check the health of remote services as described in "Monitoring GSLB Services." When monitors are bound, metric exchange does not control the state of the remote service. If a monitor is bound to a remote service and metric exchange is enabled, the monitor controls the health status. Binding the monitors to the remote service enables the NetScaler appliance to interact with a non-NetScaler load balancing device. The NetScaler can monitor non-NetScaler devices but cannot perform load balancing on them unless monitors are bound to all GSLB services and only static load balancing methods (such as the round robin, static proximity, or hash-based methods) are used.

With NetScaler release 11.1.51.x or later, to avoid unnecessary disruption of services, you can set a time delay for marking GSLB services as DOWN when a MEP connection goes DOWN.

MEP state in a high availability setup

In a high availability setup, the primary node establishes connections with the remote sites and the MEP state is not synchronized from the primary node to secondary nodes. Therefore, the MEP state in secondary node remains DOWN. When the secondary node becomes primary, it establishes MEP connections with the new GSLB site and updates the MEP state accordingly.

Enabling Site Metrics Exchange

Site metrics exchanged between the GSLB sites include the status of each load balancing, or content switching virtual server, the current number of connections, the current packet rate, and current bandwidth usage information.

The NetScaler appliance needs this information to perform load balancing between the sites. The site metric exchange interval is 1 second. A remote GSLB service must be bound to a local GSLB virtual server to enable the exchange of site metrics with the remote service.

To enable or disable site metrics exchange by using the command line interface

At a command prompt, type the following commands to enable or disable site metric exchange and verify the configuration:

```
set gslb site <siteName> -metricExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -metricExchange ENABLED
set gslb site Site-GSLB-East-Coast -metricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To enable or disable site metric exchange by using the NetScaler GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and select the site.
2. In the **Configure GSLB Site** dialog box, select the **Metric Exchange** option.

Enabling Network Metric Exchange

If your GSLB sites use the round-trip time (RTT) load balancing method, you can enable or disable the exchange of RTT information about the client's local DNS service. This information is exchanged every 5 seconds.

For details about changing the GSLB method to a method based on RTT, see "Changing the GSLB Method."

To enable or disable site metric exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and select the site.
2. In the **Configure GSLB Site** dialog box, select the **Metric Exchange** option.

Enable network metric exchange

If your GSLB sites use the round-trip time (RTT) load balancing method, you can enable or disable the exchange of RTT information about the client's local DNS service. This information is exchanged every 5 seconds.

For details about changing the GSLB method to a method based on RTT, see "GSLB Methods."

To enable or disable network metric information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable network metric information exchange and verify the configuration:

```
set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)
show gslb site <<siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -nwmetricExchange ENABLED
set gslb site Site-GSLB-East-Coast -nwmetricExchange DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To enable or disable network metric information exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**.
2. In the **Configure GSLB Site** dialog box, select the **Network Metric Exchange** option.

Configuring a time delay for the GSLB services to be marked as DOWN when a MEP connection goes DOWN

If the status of a MEP connection to a remote site changes to DOWN, the status of every GSLB service on that remote site is marked as DOWN, although the site might not actually be DOWN.

You can now set a delay to allow some time for reestablishment of the MEP connection before the site is marked as DOWN. If the MEP connection is back UP before the delay expires, the services are not affected.

For example, if you set the delay 10, the GSLB services are marked as DOWN until the MEP connection has been DOWN for 10 seconds. If the MEP connection is back UP within 10 seconds, the GSLB services remain in the UP state.

Note: This delay is applicable only to services not bound to a monitor. The delay does not affect the trigger monitors.

To set a time delay by using the command line interface

At the command prompt, type the following command:

```
set gslb parameter** - GSLBSvcStateDelayTime <sec>
<!--NeedCopy-->
```

Example:

set gslb parameter - GSLBSvcStateDelayTime 10

> **Note**
>
> In a hierarchical deployment (parent-child topology), if you configure the GSLB service on both the parent and child sites, set the GSLB parameter on both the parent and child sites. If you do not configure the GSLB service on the child site, set the GSLB parameter only on the parent site.

To set a time delay by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **GSLB** > **Change GSLB Settings**.
2. In the **GSLB Service State Delay Time (secs)** box, type the time delay in seconds.

Enable persistence information exchange

You can configure NetScaler appliance to provide persistent connections, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

You can enable or disable the exchange of persistence information at each site. This information is exchanged once every 5 seconds between NetScaler appliances participating in GSLB.

For details about configuring persistence, see Configuring Persistent Connections.

To enable or disable persistence information exchange by using the command line interface

At the command prompt, type the following commands to enable or disable persistence-information exchange and verify the configuration:

```
set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)
show gslb site** <siteName>
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-East-Coast -sessionExchange  ENABLED
set gslb site Site-GSLB-East-Coast -sessionExchange  DISABLED
show gslb site Site-GSLB-East-Coast
<!--NeedCopy-->
```

To enable or disable persistence information exchange by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and double-click the site.
2. In the **Configure GSLB Site** dialog box, select or clear the **Persistence Session Entry Exchange** check box.

Configuring Site-to-Site Communication

September 21, 2020

GSLB site-to-site communication is between the remote procedure call (RPC) nodes that are associated with the communicating sites. A master GSLB site establishes connections with slave sites to synchronize GSLB configuration information and to exchange site metrics.

An RPC node is created automatically when a GSLB site is created, and is assigned an internally generated user name and password. The NetScaler appliance uses this user name and password to authenticate itself to remote GSLB sites during connection establishment. No configuration steps are necessary for an RPC node, but you can specify a password of your choice, enhance security by encrypting the information that GSLB sites exchange, and specify a source IP address for the RPC node.

The appliance needs a NetScaler owned IP address to use as the source IP address when communicating with other GSLB sites. By default, the RPC nodes use a subnet IP (SNIP) address, but you might want to specify an IP address of your choice.

The following topics describe the behavior and configuration of RPC nodes on the NetScaler appliance:

Changing the password of an RPC node

Citrix recommends you to secure the communication between sites in your GSLB setup by changing the password of each RPC node. After you change the password for the RPC node of the local site, you must manually propagate the change to the RPC node at each of the remote sites.

The password is stored in encrypted form. You can verify that the password has changed by using the show rpcNode command to compare the encrypted form of the password before and after the change.

To change the password of an RPC node by using the command line interface

At the command line, type the following commands to change the password of an RPC node:

```
set ns rpcNode <IPAddress> {
 -password }

show ns rpcNode
<!--NeedCopy-->
```

Example:

```
> set rpcNode 192.0.2.4 -password mypassword
 Done
> show rpcNode
.
.
.
2)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
    SrcIP:  *                    Secure:  OFF
Done
>

<!--NeedCopy-->
```

To unset the password of an RPC node by using the command line interface

To unset the password of an RPC node by using the CLI, type the unset rpcNode command, the IP address of the RPC node, and the password parameter, without a value.

To change the password of an RPC node by using the configuration utility

Navigate to System > Network > RPC, select the RPC node, and change the password.

Encrypt the exchange of site metrics

You can secure the information that is exchanged between GSLB sites by setting the secure option for the RPC nodes in the GSLB setup. With the secure option set, the NetScaler appliance encrypts all communication sent from the node to other RPC nodes.

To encrypt the exchange of site metrics by using the command line interface

At the command prompt, type the following commands to encrypt the exchange of site metrics and verify the configuration:

```
set ns rpcNode <IPAddress> [-secure ( YES | NO )]
show rpcNode
<!--NeedCopy-->
```

Example:

```
> set rpcNode 192.0.2.4 -secure YES
 Done
>
> show rpcNode
.
.
.
3)  IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:
    192.0.2.3     Secure:  ON
 Done
>
<!--NeedCopy-->
```

To unset the secure parameter by using the command line interface

To unset the secure parameter by using the CLI, type the unset rpcNode command, the IP address of the RPC node, and the secure parameter, without a value.

To encrypt the exchange of site metrics by using the NetScaler configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. Select the **Secure** option, and click **OK**.

Configure source IP address for an RPC node

By default, the NetScaler appliance uses a NetScaler owned subnet IP (SNIP) address as the source IP address for an RPC node, but you can configure the appliance to use a specific SNIP address. If a SNIP address is not available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NSIP address or a virtual IP (VIP) address as the source IP address for an RPC node. A VIP address can be used as the source IP address of an RPC node only if the RPC node is a remote node. If you configure a VIP address as the source IP address and remove the VIP address, the appliance uses a SNIP address.

> **Note**
>
> From NetScaler 11.0.64.x release onwards, you can configure the appliance to use GSLB Site IP address as the source IP address for an RPC node.

To specify a source IP address for an RPC node by using the command line interface

At the command prompt, type the following commands to change the source IP address for an RPC node and verify the configuration:

```
set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
show ns rpcNode
<!--NeedCopy-->
```

Example:

```
set rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
show rpcNode
<!--NeedCopy-->
```

```
IPAddress: 192.0.2.4 Password: d336004164d4352ce39e SrcIP: 192.0.2.3
    Secure: OFF
Done
<!--NeedCopy-->
```

To unset the source IP address parameter by using the command line interface

To unset the source IP address parameter by using the CLI, type the unset rpcNodecommand, the IP address of the RPC node, and the srcIP parameter, without a value.

To specify a source IP address for an RPC node by using the NetScaler configuration utility

1. Navigate to System > Network > RPC and double-click a RPC node.
2. In the Source IP Address field, enter the IP address that you want the RPC node to use as the source IP address and click OK.

> **Important**
>
> The source IP address cannot be synchronized across the sites participating in GSLB because the source IP address for a RPC node is specific to each NetScaler appliance. Therefore, after you force a synchronization (using the sync gslb config –forceSync command or by selecting the ForceSync option in the GUI), you have to manually change the source IP addresses on the other NetScaler appliances.

Customize your GSLB configuration

September 21, 2020

Once your basic GSLB configuration is operational, you can customize it by modifying the bandwidth of a GSLB service, configuring CNAME based GSLB services, static proximity, dynamic RTT, persistent connections, or dynamic weights for services, or changing the GSLB Method.

You can also configure monitoring for GSLB services to determine their states.

These settings depend on your network deployment and the types of clients you expect to connect to your servers.

Modify maximum connections or maximum bandwidth for a GSLB service

You can restrict the number of new clients that can simultaneously connect to a load balancing or content switching virtual server by configuring the maximum number of clients and/or the maximum bandwidth for the GSLB service that represents the virtual server.

To modify the maximum clients or bandwidth of a GSLB service by using the command line interface

At the command prompt, type the following command to modify the maximum number of client connections or the maximum bandwidth of a GSLB service and verify the configuration:

```
set gslb service <serviceName> [-maxClients <positive_integer>] [-
    maxBandwidth <positive_integer>]
show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
set glsb service Service-GSLB-1  - maxBandwidth 100  - maxClients 100
show gslb service Service-GSLB-1
<!--NeedCopy-->
```

To modify the maximum clients or bandwidth of a GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**, and double-click a service.
2. Click in the **Other Settings** section and set the following parameters:
    - Max Clients—maxClients
    - Max Bandwidth—maxBandwidth

Create CNAME-based GSLB services

To configure a GSLB service, you can use the IP address of the server or a canonical name of the server. If you want to run multiple services (like an FTP and a Web server, each running on different ports) from a single IP address or run multiple HTTP services on the same port, with different names, on the same physical host, you can use canonical names (CNAMES) for the services.

For example, you can have two entries in DNS as ftp.example.com and www.example.com for FTP services and HTTP services on the same domain, example.com. CNAME-based GSLB services are useful in a multilevel domain resolver configuration or in multilevel domain load balancing. Configuring a CNAME-based GSLB service can also help if the IP address of the physical server is likely to change.

If you configure CNAME-based GSLB services for a GSLB domain, when a query is sent for the GSLB domain, the NetScaler appliance provides a CNAME instead of an IP address. If the A record for this CNAME record is not configured, the client must query the CNAME domain for the IP address. If the A record for this CNAME record is configured, the NetScaler appliance provides the CNAME with the corresponding A record (IP address). The NetScaler appliance handles the final resolution of the DNS query, as determined by the GSLB method. The CNAME records can be maintained on a different NetScaler appliance or on a third-party system.

In an IP-address-based GSLB service, the state of a service is determined by the state of the server that it represents. However, a CNAME-based GSLB service has its state set to UP by default; the virtual server IP (VIP) address or metric exchange protocol (MEP) are not used for determining its state. If a desktop-based monitor is bound to a CNAME-based GSLB service, the state of the service is determined according to the result of the monitor probes.

You can bind a CNAME-based GSLB service only to a GSLB virtual server that has the DNS Record Type as CNAME. Also, a NetScaler appliance can contain at most one GSLB service with a given CNAME entry.

The following are some of the features supported for a CNAME-based GSLB service:

- GSLB-policy based site affinity is supported, with the CNAME as the preferred location.
- Source IP persistence is supported. The persistency entry contains the CNAME information instead of the IP address and port of the selected service.

The following are the limitations of CNAME-based GSLB services:

- Site persistence is not supported, because the service referenced by a CNAME can be present at any third-party location.
- Multiple-IP-address response is not supported because one domain cannot have multiple CNAME entries.
- Source IP Hash and Round Robin are the only load balancing methods supported. The Static Proximity method is not supported because a CNAME is not associated with an IP address and static proximity can be maintained only according to the IP addresses.

Note: The
Empty-Down-Response feature should be enabled on the GSLB virtual server to which you bind the CNAME-based GSLB service. If you enable the
Empty-Down-Response feature, when a GSLB virtual server is DOWN or disabled, the response to a DNS query, for the domains bound to this virtual server, contains an empty record without any IP addresses, instead of an error code.

To create a CNAME-based GSLB service by using the command line interface

At the command prompt, type:

```
add gslb service <serviceName> -cnameEntry <string> -siteName <string>
<!--NeedCopy-->
```

Example:

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -siteName Site-GSLB-East-Coast
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -siteName Site-GSLB-West-Coast
<!--NeedCopy-->
```

To create a CNAME-based GSLB service by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Services**.
2. Create a service, and set the **Type to Canonical Name Based**.

Configure transition Out-Of-Service State (TROFS) in GSLB

When you configure persistence on a GSLB virtual server to which a service is bound, the service continues to serve requests from the client even after it is disabled, accepting new requests or connections only to honor persistence. After a configured period of time, known as the graceful shutdown period, no new requests or connections are directed to the service, and all of the existing connections are closed.

When disabling a service, you can specify a graceful shutdown period, in seconds, by using the delay argument. During the graceful shutdown period, if the service is bound to a virtual server, its state appears as Out of Service.

Configure dynamic weights for services

In a typical network, there are servers that have a higher capacity for traffic than others. However, with a regular load balancing configuration, the load is evenly distributed across all services even though different services represent servers with different capacities.

To optimize your GSLB resources, you can configure dynamic weights on a GSLB virtual server. The dynamic weights can be based on either the total number of services bound to the virtual server or the sum of the weights of the individual services bound to the virtual server. Traffic distribution is then based on the weights configured for the services.

When dynamic weights are configured on the GSLB virtual server, requests are distributed according to the load balancing method, the weight of the GSLB service, and the dynamic weight. The product of the weight of the GSLB service and the dynamic weight is known as the cumulative weight. Therefore, when dynamic weight is configured on the GSLB virtual server, requests are distributed on the basis of the load balancing method and the cumulative weight.

When dynamic weight for a virtual server is disabled, the numerical value is set to 1. This ensures that the cumulative weight is a non-zero integer at all times.

Dynamic weight can be based on the total number of active services bound to load balancing virtual servers or on the weights assigned to the services.

Consider a configuration with two GSLB sites configured for a domain and each site has two services that can serve the client. If a service at either site goes down, the other server in that site has to handle twice as much traffic as a service at the other site. If dynamic weight is based on the number of active services, the site with both services active has twice the weight of the site with one service down and therefore receives twice as much traffic.

Alternatively, consider a configuration in which the services at the first site represent servers that are twice as powerful as servers at the second site. If dynamic weight is based on the weights assigned to the services, twice as much traffic can be sent to the first site as to the second.

Note: For details on assigning weights to load balancing services, see “
Assigning Weights to Services”.

As an illustration of how dynamic weight is calculated, consider a GSLB virtual server that has a GSLB service bound to it. The GSLB service represents a load balancing virtual server that in turn has two services bound to it. The weight assigned to the GSLB service is 3. The weights assigned to the two services are 1 and 2 respectively. In this example, when dynamic weight is set to:

- **Disabled**:The cumulative weight of the GSLB virtual server is the product of the dynamic weight (disabled = 1) and the weight of the GSLB service (3), so the cumulative weight is 3.
- **SERVICECOUNT**: The count is the sum of the number of services bound to the load balancing virtual servers corresponding to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.
- **SERVICEWEIGHT**: The dynamic weight is the sum of the number of services bound to the GSLB service (2), and the cumulative weight is the product of the dynamic weight (2) and the weight of the GSLB service (3), which is 6.

Note: Dynamic weights are not applicable when content switching virtual servers are configured.

To configure a GSLB virtual server to use dynamic weights by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight SERVICECOUNT | SERVICEWEIGHT
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
<!--NeedCopy-->
```

To set GSLB virtual server to use dynamic weights by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers, double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).
2. Click the **Method** section and, from the **Dynamic Weight** drop-down list, select **SERVICECOUNT** or **SERVICEWEIGHT**.

Changing the GSLB methods

September 21, 2020

Unlike traditional DNS servers that simply respond with the IP addresses of the configured servers, a NetScaler appliance configured for GSLB responds with the IP addresses of the services, as determined by the configured GSLB method. By default, the GSLB virtual server is set to the least connection method. If all GSLB services are down, the appliance responds with the IP addresses of all the configured GSLB services.

GSLB methods are algorithms that the GSLB virtual server uses to select the best-performing GSLB service. After the host name in the Web address is resolved, the client sends traffic directly to the resolved service IP address.

The NetScaler appliance provides the following GSLB methods:

- Round Robin
- Least Connections
- Least Response Time
- Least Bandwidth
- Least Packets
- Source IP Hash
- Custom Load
- Round Trip Time (RTT)
- Static Proximity

For GSLB methods to work with a remote site, either MEP must be enabled or explicit monitors must be bound to the remote services. If MEP is disabled, RTT, Least Connections, Least Bandwidth, Least Packets and Least Response Time methods default to Round Robin.

The Static Proximity and RTT load balancing methods are specific to GSLB.

Specifying a GSLB method other than static proximity or dynamic RTT

For information about the Round Robin, Least Connections, Least Response Time, Least Bandwidth, Least Packets, Source IP Hash, or Custom Load method, see “
Load Balancing.”

To change the GSLB method by using the CLI

At the command prompt, type:

```
set gslb vserver <name> -lbMethod GSLBMethod
```

```
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
<!--NeedCopy-->
```

To change the GSLB method by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers**.
2. In the details pane, select a GSLB virtual server and click **Open**.
3. In the Configure GSLB Virtual Server dialog box, on the Method and Persistence tab, under Method, select a method from the Choose Method list.
4. Click **OK**, and verify that the method you selected appears under Details at the bottom of the screen.

Configuring Static Proximity

September 21, 2020

The static proximity method for GSLB uses an IP-address based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

For the static proximity method to work, you must either configure the NetScaler appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the GSLB method.

This document includes the following information:

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers
- Specifying the Proximity Method
- Synchronizing GSLB Static Proximity Database

Add a location file to create a static proximity database

September 21, 2020

A static proximity database is a UNIX-based ASCII file. Entries added to this database from a location file are called static entries. Only one location file can be loaded on a NetScaler appliance. Adding a new location file overrides the existing file. The number of entries in the static proximity database is limited by the configured memory in the NetScaler appliance.

The static proximity database can be created in the default format or in a format derived from commercially configured third party databases (such as www.maxmind.com and www.ip2location.com).

The NetScaler appliance includes the following two IP geolocation database files. These are GeoLite2 files, published by MaxMind.

- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4
- Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6

These database files are available in a format supported by the NetScaler appliance in the directory /var/netscaler/inbuilt_db.

You can use these IP geolocation databases as the location file for the static proximity based GSLB method, or in location based policies.

These databases vary in the details they provide. There is no strict enforcement of the database file format, except that the default file has format tags. The database files are ASCII files that use a comma as the field delimiter. There are differences in the structure of fields and the representation of IP addresses in the locations.

The format parameter describes the structure of the file to the NetScaler appliance. Specifying an incorrect value for the format option can corrupt the internal data.

> **Note**
>
> - After an upgrade, if the /var/netscaler/inbuilt_db/ directory contains the da tabase file (Citrix_Netscaler_InBuilt_GeoIP_DB.csv) from the earlierNetScaler software versions, the file is retained.
> - The default location of the database file is /var/netscaler/locdb, and on a high availability (HA) setup, an identical copy of the file must be present in the same location on both NetScaler appliances.
> - If the location file is stored in a location other than the default location, then specify the path of the location file.
> - For admin partitions, the default path is: /var/partitions/<`partitionName`>/netscaler/locdb.
> - Some databases provide short country names according to ISO-3166 and long country names as well. The NetScaler uses short names when storing and matching qualifiers.

> - To create a static proximity database, log on to the UNIX shell of the NetScaler appliance and use an editor to create a file with the location details in one of the NetScaler-supported formats.

To add a static location file by using the CLI

At the command prompt, type:

```
add locationFile <locationFile> [-format <format>]
-  show locationFile
<!--NeedCopy-->
```

Example:

```
add locationFile /var/nsmap/locdb/nsgeo1.0 -format netscaler
Done

show locationFile
Location File: /var/nsmap/locdb/nsgeo1.0
Format: netscaler
 Done
>
<!--NeedCopy-->
```

Example:

```
add locationFile /var/netscaler/inbuilt_db/
    Citrix_Netscaler_InBuilt_GeoIP_DB_IPv4 -format netscaler

add locationFile6 /var/netscaler/inbuilt_db/
    Citrix_Netscaler_InBuilt_GeoIP_DB_IPv6 -format netscaler
<!--NeedCopy-->
```

To add a static location file by using the GUI

1. Navigate to **AppExpert** > **Location**, click the **Static Database** tab.
2. Click **Add** to add a static location file.

You can view an imported location file database by using the View Database dialog box in the configuration utility. There is no CLI equivalent.

To view a static location file by using the GUI

1. Navigate to **AppExpert** > **Location**, click the **Static Database** tab.

2. Select a static location file, and from the **Action** list, click **View Database**.

To convert a location file into the NetScaler format

By default, when you add a location file, it is saved in the NetScaler format. You can convert a location file of other formats into the NetScaler format.

Note: The nsmap option can be accessed only from the command line interface. The conversion is possible only into the NetScaler format.

To convert the static database format, At the CLI prompt, type the following command:

```
nsmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
<!--NeedCopy-->
```

Example:

```
 nsmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.
    csv
<!--NeedCopy-->
```

Add a third-party static database file on a NetScaler appliance

Perform the following steps to add a third-party static database file on a NetScaler appliance.

1. Obtain the location database file from a third-party vendor, such as www.maxmind.com or www.ip2location.com.
2. Copy the location database file to the NetScaler appliance using the WinSCP utility.

   > **Note**
   >
   > The default location of the database file on the appliance is **/var/netscaler/locdb**.

3. Execute the following command to add a static location file:

   ```
   add location file <locationfile Name> -format LocationFormat
   <!--NeedCopy-->
   ```

4. Execute the following command to ensure that the location database is loaded:

   ```
   show location parameter
   <!--NeedCopy-->
   ```

   This command displays the parameters, such as number of static entries. If the database is not loaded correctly, this command also displays an error message. A maximum of 3M-1 (3 million minus one) entries can be loaded.

5. Execute the following command to view the location of the GSLB site:

```
show gslb service
<!--NeedCopy-->
```

> **Note**
>
> - If the database is loaded correctly, the location of the GSLB sites is automatically populated in the database.
> - You can specify only one location file in the configuration on the appliance.
> - If the appliances are in a high availability setup, then one appliance must copy the database from the other appliance.
> - If no match is found for an incoming IP address, the request is processed using the Round Robin method.

6. Execute the following command to configure the GSLB method on the appliance:

```
set gslb vserver GSLBVserverName -lbMethod MethodType
<!--NeedCopy-->
```

Add custom entries to a static proximity database

September 21, 2020

Custom entries take precedence over static entries in the proximity database. You can add a maximum of 500 custom entries. For a custom entry, denote all omitted qualifiers with an asterisk (*) and, if qualifiers have a period or space in the name, enclose the parameter in double quotation marks. The first 31 characters are evaluated for each qualifier. You can also provide the longitude and latitude of the geographical location of the IP address-range for selecting a service with the static proximity GSLB method.

To add custom entries by using the command line interface

At the command prompt, type the following commands to add a custom entry to the static proximity database and verify the configuration:

```
add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer
   >[-latitude <integer>]]
show location
<!--NeedCopy-->
```

Example:

```
>add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
<!--NeedCopy-->
```

```
>show location
<!--NeedCopy-->
```

Parameters for adding custom entries

- IPfrom

  First IP address in the range, in dotted decimal notation. This is a mandatory argument.

- IPto

  Last IP address in the range, in dotted decimal notation. This is a mandatory argument.

- preferredLocation

  String of qualifiers, in dotted notation, describing the geographical location of the IP address range. Each qualifier is more specific than the one that precedes it, as in continent.country.region.city.isp.organization. For example,"NA.US.CA.San Jose.ATT.citrix".
  Note: A qualifier that includes a dot (.) or space ( ) must be enclosed in double quotation marks.
  This is a mandatory argument. Maximum Length: 197

- longitude

  Numerical value, in degrees, specifying the longitude of the geographical location of the IP address-range.
  Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.
  Maximum value: 180

- latitude

  Numerical value, in degrees, specifying the latitude of the geographical location of the IP address-range.
  Note: Longitude and latitude parameters are used for selecting a service with the static proximity GSLB method. If they are not specified, selection is based on the qualifiers specified for the location.
  Maximum value: 180

To add custom entries by using the configuration utility

Navigate to **AppExpert** > **Location**, click the **Custom Entries** tab, and add the custom entries.

Set location qualifiers

September 21, 2020

The database used to implement static proximity contains the location of the GSLB sites. Each location contains an IP address range and up to six qualifiers for that range. The qualifiers are literal strings and are compared in a prescribed order at run time. Every location must have at least one qualifier. The meaning of the qualifiers (context) is defined by the qualifier labels, which are user defined. NetScaler has two built-in contexts:

Geographic context, which has the following qualifier labels:

- Qualifier 1 – "Continent"
- Qualifier 2 – "Country"
- Qualifier 3 – "State"
- Qualifier 4 – "City"
- Qualifier 5 – "ISP"
- Qualifier 6 – "Organization"

Custom entries, which have the following qualifier labels:

- Qualifier 1 – "Qualifier 1"
- Qualifier 2 – "Qualifier 2"
- Qualifier 3 – "Qualifier 3"
- Qualifier 4 – "Qualifier 4"
- Qualifier 5 – "Qualifier 5"
- Qualifier 6 – "Qualifier 6"

If the geographic context is set with no Continent qualifier, Continent is derived from Country. Even the built-in qualifier labels are based on the context, and the labels can be changed. These qualifier labels specify the locations mapped with the IP addresses used to make static proximity decisions.

To perform a static proximity-based decision, the NetScaler appliance compares the location attributes (qualifiers) derived from the IP address of the local DNS server resolver with the location attributes of the participating sites. If only one site matches, the appliance returns the IP address of that site. If there are multiple matches, the site selected is the result of a round robin on the matching GSLB sites. If there is no match, the site selected is a result of a round robin on all configured sites. A site that does not have any qualifiers is considered a match.

To set the location qualifiers by using the command line interface

At the command prompt, type:

```
set locationparameter -context <context> -q1label <string> [-q2label <
    string>] [-q3label <string>] [-q4label <string>] [-q5label <string>]
     [-q6label <string>]
<!--NeedCopy-->
```

Example:

```
set locationparameter -context custom -q1label asia
<!--NeedCopy-->
```

To set the location qualifiers by using the configuration utility

1. Navigate to **AppExpert** > **Location**.
2. From the **Action** list, click **Location Parameters** and set the location qualifiers.

Specify proximity method

September 21, 2020

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

```
set gslb vserver <name> -lbMethod STATICPROXIMITY
show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
<!--NeedCopy-->
```

To specify static proximity by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server.

2. Click the **Method** section and from the **Choose Method** drop-down list, select **STATICPROXIMITY**.

Synchronize GSLB static proximity database

September 21, 2020

Synchronizing a global server load balancing (GSLB) static proximity database requires that one of the sites be identified as the master GSLB node. Any site in the topology can be designated as the master node. The rest of the GSLB nodes are automatically designated as slave nodes.

Synchronizing GSLB static proximity databases synchronizes the files in the /var/netscaler/locdb directory across the slave nodes. During the synchronization process, the master node fetches the running configuration from each of the slave nodes and compares it to the configuration on the master node. The master GSLB node uses the rsync program to synchronize the static proximity database across the slave nodes. To speed up the synchronization process, the rsync program makes only enough changes to eliminate the differences between the two files. The synchronization process cannot be rolled back.

The following example synchronizes Site2, which is a slave site, to master site Site1. The administrator enters the **sync gslb config** command on Site1:

```
sync gslb config -nowarn
Sync Time: Feb 24 2014 14:56:16
Retrieving local site info: ok
Retrieving all participating gslb sites info:
0 bytes in 0 blocks
ok
site1[Master]:
        Getting Config: ok
site2[Slave]:
        Syncing gslb static proximity database: ok
        Getting Config: ok
        Comparing config: ok
        Applying changes: ok
 Done
<!--NeedCopy-->
```

Configuring the Dynamic Method (RTT)

November 13, 2020

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The appliance then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value.

When a client's DNS request for a domain comes to the NetScaler appliance configured as the authoritative DNS for that domain, the appliance uses the RTT value to select the IP address of the best performing site to send it as a response to the DNS request.

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), UDP, and TCP to gather the RTT metrics for connections between the local DNS server and participating sites. The appliance first sends a ping probe to determine the RTT. If the ping probe fails, a DNS UDP probe is used. If that probe also fails, the appliance uses a DNS TCP probe.

These mechanisms are represented on the NetScaler appliance as Load Balancing Monitors and are easily identified due to their use of the "ldns" prefix. The three monitors, in their default order, are:

- ldns-ping
- ldns-dns
- ldns-tcp

These monitors are built in to the appliance and are set to safe defaults, but may be customized just like any other monitor on the appliance.

The default order may also be changed by setting it explicitly as a GSLB parameter. For example, to set the order to be the DNS UDP query followed by the PING and then TCP, type the following command:

set gslb parameter -ldnsprobeOrder DNS PING TCP

Unless they have been customized, the NetScaler appliance performs UDP and TCP probing on port 53, however unlike regular load balancing monitors the probes need not be successful in order to provide valid RTT information. ICMP port unavailable messages, TCP Resets and DNS error responses, which would usually constitute a failure are all acceptable for calculating the RTT value.

Once the RTT data has been compiled, the appliance uses the proprietary metrics exchange protocol (MEP) to exchange RTT values between participating sites. After calculating RTT metrics, the appliance sorts the RTT values to identify the data center with the best (smallest) RTT metric."

If RTT information is not available (for example, when a client's local DNS server accesses the site for the first time), the NetScaler appliance selects a site by using the round robin method and directs the client to the site.

To configure the dynamic method, you configure the site's GSLB virtual server for dynamic RTT. You can also set the interval at which local DNS servers are probed to a value other than the default.

Configure a GSLB virtual server for dynamic RTT

To configure a GSLB virtual server for dynamic RTT, you specify the RTT load balancing method.

The NetScaler appliance regularly validates the timing information for a given local server. If a change in latency exceeds the configured tolerance factor, the appliance updates its database with the new timing information and sends the new value to other GSLB sites by performing a MEP exchange. The default tolerance factor is 5 milliseconds (ms).

The RTT tolerance factor must be the same throughout the GSLB domain. If you change it for a site, you must configure identical RTT tolerance factors on all NetScaler appliances deployed in the GSLB domain.

To configure a GSLB virtual server for dynamic RTT by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -lbMethod RTT -tolerance <value>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
<!--NeedCopy-->
```

To configure a GSLB virtual server for dynamic RTT by using the configuration utility

Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the virtual server.

Set the probing interval of local DNS servers

The NetScaler appliance uses different mechanisms, such as ICMP echo request / reply (PING), TCP, and UDP to obtain RTT metrics for connections between the local DNS server and participating GSLB sites. By default, the appliance uses a ping monitor and probes the local DNS server every 5 seconds. The appliance then waits 2 seconds for the response and, if a response is not received in that time, it uses the TCP DNS monitor for probing.

However, you can modify the time interval for probing the local DNS server to accommodate your configuration.

To modify the probing interval by using the command line interface

At the command prompt, type:

```
set lb monitor <monitorName> <type> -interval <integer> <units> -
    resptimeout <integer> <units>
<!--NeedCopy-->
```

Example:

```
set lb monitor ldns-tcp LDNS-TCP -interval 10 sec -resptimeout 5 sec
<!--NeedCopy-->
```

To modify the probing interval by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Monitors**, and double-click the monitor that you want to modify (for example, PING)

How to configure persistence in GSLB

January 4, 2021

Persistence ensures that a series of client requests for a particular domain name is sent to the same data center instead of being load balanced. If persistence is configured for a particular domain, it takes precedence over the configured GSLB method. You can use persistence for deployments where an information related to a client transaction is stored locally on an instance, which has served the initial requests. For example, the deployments for e-commerce that uses a shopping cart, where the server needs to maintain the state of the connection to track the transaction. The NetScaler appliance selects a data center to process a client request. With persistence enabled, it forwards the same IP address of the selected data center for all subsequent Domain Name System (DNS) requests. If a persistence session points to a data center that is DOWN, the NetScaler appliance uses the configured GSLB method to select a new data center. It then becomes persistent for subsequent requests from the client.
For persistence in GSLB, the same set of persistence identifiers (persistID) must be configured on the GSLB virtual servers in all data centers. The GSLB module uses the persistence identifier to uniquely identify a GSLB virtual server. When Source IP persistence is enabled on the GSLB virtual server, the persistence sessions are also exchanged as part of the metrics exchange. For the NetScaler appliance to support persistence across sites, persistence related configuration must be done on all the participating GSLB sites. Citrix recommends persistence in GSLB for stateful applications, which requires clients to reconnect to the same application instance for the subsequent requests.

You can achieve persistence in GSLB by the following ways:

- Persistence on GSLB virtual server
- Site persistence on GSLB services

Persistence on GSLB virtual server

Persistence on GSLB virtual server is used during the DNS requests. The Source IP address of the DNS request is used to create persistence session between the client and the data center. DNS clients are generally the Local DNS (LDNS) or DNS gateways proxying a set of clients sitting behind them (in ISPs). Persistence on a GSLB virtual server is application protocol agnostic.
In general, multiple DNS gateways or Local Domain Name Servers (LDNS) are configured in the client network. Citrix recommends you to configure an appropriate persistence mask because for the subsequent DNS requests, irrespective of the upstream LDNS devices used to connect to the NetScaler appliance, the client is able to persist to the same data center, which had served the earlier requests. After the persistence session is created for an LDNS IP address, all the end clients connecting using that LDNS are given the same data center IP address.

Site persistence on GSLB services

Site persistence becomes effective while processing the application requests. Site persistence works only for HTTP and HTTPS traffic because the persistency is achieved using HTTP cookie. As cookies are maintained on HTTP clients (browsers), it gives visibility into the clients sitting behind the DNS gateways. When you use cookies to achieve persistency for clients, no resources are consumed on the NetScaler appliance for each incoming client. When you bring a GSLB service DOWN with a delay time, the service goes into the transition to out of service (TROFS) state. Persistence is supported as long as the service is in the UP or TROFS state. That is, if the same client sends a request for the same service within the specified delay time after a service is marked TROFS, the same GSLB site (data center) services the request.

If you access an application through an alias, ensure that the CNAME record is also configured on the Citrix ADC appliance. In a parent-child topology, site persistence does not work when you access an application through an alias.

> Note
>
> If the connection proxy is specified as the site persistence method and you also want to configure persistence on LB virtual servers, source IP persistence is not recommended. When the connection is proxied, an IP address owned by the NetScaler appliance is used, and not the actual IP address of the client.
> Configure an appropriate persistence, which does not use source IP of the HTTP(S) request to identify the client, for example, cookie persistence or rule-based persistence.

Configure persistence based on source IP address

If source IP persistence is configured on GSLB virtual server, persistence sessions are created for the source IP address of the DNS request. Depending on the Extended Client Subnet (ECS) feature, the source IP address of the DNS request is taken from any of the following:

- The source IP in the IP header of the incoming DNS request packet
- The ECS option in the DNS request For more information on ECS, see Use the EDNS0 client subnet option for Global Server Load Balancing.

Persistence sessions for a client last until the persistence timeout. After the timeout period expires, existing persistence sessions are cleared. For subsequent requests, a new GSLB decision is made and a different GSLB service IP address might be selected.
The source IP persistence on GSLB virtual server and site persistence on GSLB service complements each other. If source IP persistence is disabled on GSLB virtual server, the GSLB virtual server chooses a different GSLB service each time the DNS tries to do the resolution. The client also connects to a different GSLB service and the data center which receives the application request proxy the connection to the data center which served the client first. This might add some latency. So by enabling source IP persistence on GSLB virtual server can avoid frequent such multiple hops for application requests. If the source IP persistence session has expired and the client reconnects after that, the site persistence connects the client back to the data center, which had served the client initially. Also, if the client connects back through a DNS gateway, which does not fall within the persistence mask range configured, then as well site persistence helps clients stick to the data center that served the first request.

To configure persistence based on source IP address by using the CLI

At the command prompt, type:

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId <positive_integer> [-persistMask <netmask>]  - [timeout <mins>]
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -persistenceId 23 -persistMask 255.255.255.255  - timeout 2
<!--NeedCopy-->
```

To configure persistence based on source IP address by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Virtual Servers** and double-click the GSLB virtual server whose method you want to change (for example, vserver-GSLB-1).

2. Click the **Persistence** section and, from the **Persistence** drop-down list, select **SOURCEIP** and set the following parameters:
   - Persistence Id—persistenceID
   - Time-out—timeout
   - IPv4 Netmask or IPv6 Mask length—persistMask

Configure site persistence based on HTTP cookies

Site persistence is achieved using HTTP cookies (known as a "site cookie") to reconnect the client to the same server. When the GSLB appliance responds to a client DNS request by sending the IP address of the selected GSLB site, the client sends an HTTP request to that GSLB site. Application endpoint in that GSLB site adds a site cookie to the HTTP header, and site persistence is in effect.
If the client sends a DNS query after the client cache is expired, the DNS request might be directed to a different GSLB site. The new GSLB site uses the site cookie present in the client request header to implement persistence. Site persistence feature becomes active under the following conditions:

- When the domain name in the host header matches one of the GSLB domains
- When site persistence is enabled on the GSLB service that represents the virtual server receiving the application traffic.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. If the GSLB service pointed by the cookie is DOWN or removed from the GLSB configuration, then the virtual server that receives the traffic continues to process the traffic. The cookie expiration is based on the cookie timeout configured on the NetScaler appliance. If the virtual server names are not identical on all the sites, you must use the persistence identifier. Cookies inserted are compliant with RFC 2109.

NetScaler supports two types of site persistence:

- Connection Proxy
- HTTP redirect

Connection Proxy

In the Connection Proxy mode of site persistence, the data center that receives the subsequent application request performs the following tasks to establish a connection:

1. Creates a connection to the GSLB site that inserted the site cookie.

2. Proxies the client request to the original site.

   > **Note:**
   >
   > The proxy server establishes connection with the original site using the following details:

> - The SNIP of the new site is the source IP address.
> - The GSLB service public IP address of the original site is the destination IP address.
> - An ephemeral port is the source port and GSLB service port is the destination port.
> - Uses either HTTP or HTTPS protocols depending on the GSLB service type.

3. Receives a response from the original GSLB site.
4. Relays that response back to the client.
5. Closes the connection.

HTTP redirect

If the GSLB configuration uses HTTP redirect persistence, the new site redirects the request to the site that originally inserted the cookie. Domain name in the redirect URL is the site domain. Ensure that both cookies and SSL certificates are applicable to both GSLB domain and site domain. To apply cookies for both GSLB and site domain, the cookie domain must be the site to GSLB domain. To apply SSL certificates to both GSLB and site domain, certificate bound to the SSL virtual server must be a wildcard certificate.

Connection proxy occurs when the following conditions are satisfied:

- Requests are sent for a domain participating in GSLB. The domain is obtained from the URL/Host header.
- The local GSLB service has connection proxy enabled.
- The request includes a valid cookie that contains the IP address of an active remote GSLB service.

> **Note**
>
> In a GSLB parent-child configuration, the connection proxy works as intended even when a GSLB service is not configured on a child site. However, if you have extra configuration such as client authentication, client IP address insertion, or other SSL-specific requirement, you must add an explicit GSLB service on the site and configure it accordingly.
>
> For more information about parent-child topology, see Parent-Child Topology Deployment Using the MEP Protocol.

To set persistence based on HTTP cookies by using the CLI

At the command prompt, type:

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-
    sitePrefix <prefix>] | HTTPRedirect -sitePrefix <prefix>)
<!--NeedCopy-->
```

Example:

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
<!--NeedCopy-->
```

To set persistence based on cookies by using the GUI

1. Navigate to **Traffic Management** > **GSLB** > **Services** and select the service that you want to configure for site persistence (for example, service-GSLB-1).
2. Click the **Site Persistence** section and set persistence based on cookies.

Override static proximity behavior by configuring preferred locations

September 21, 2020

You might want to direct traffic from a local DNS (LDNS) server or network to a GSLB service other than the GSLB service that the static proximity method selects for that traffic. That is, you have a *preferred location* for that traffic. To override the static proximity method with preferred locations, you can do the following:

1. Configure a DNS action that consists of a list of preferred locations. For more information about configuring a DNS action, see Configuring a DNS Action.
2. Configure a DNS policy to identify the traffic arriving from the LDNS server or network for which you want to override static proximity, and apply the action in the policy.
3. Bind the policy to the global request bind point.

In the DNS action, you can configure a list of up to 8 preferred locations. The locations must be provided in the dotted qualifier notation, which is the notation in which you add custom locations to the static proximity database. The locations can include wildcards for qualifiers that you want to omit. For information about the dotted qualifier notation for locations, see Adding Custom Entries to a Static Proximity Database. When entering the preferred locations, you must enter them in the descending order of priority.

When a policy evaluates to TRUE, the NetScaler appliance matches the preferred locations, in priority order, with the locations of GSLB services. Matches are of the following two types:

- If all the non-wildcard qualifiers in a preferred location match the corresponding qualifiers in the location of a GSLB service, the match is considered a perfect match. For example, a GSLB service location of _.UK._* or Europe.UK.. is a perfect match for the preferred location _.UK._*.

- If only a subset of the non-wildcard qualifiers match, the match is considered a partial match. For example, a GSLB service location of Europe.EG is a partial match for the preferred location Europe.UK.

When a DNS policy evaluates to
TRUE, the following algorithm is used to select a GSLB service:

1. The appliance evaluates the preferred location that has the highest priority and moves down the priority order until a perfect match is found between a preferred location and the location of a GSLB service.

   If a perfect match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple perfect matches are found (which can happen when one or more wildcards are used in a preferred location), the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

2. If a perfect match is not found for any of the preferred locations, the appliance returns to the preferred location that has the highest priority and moves down the priority order until a partial match is found between a preferred location and the location of a GSLB service.

   If a partial match is found, the appliance checks whether the corresponding GSLB service is up. If it is up, it returns the IP address of the GSLB service in the DNS response. If multiple partial matches are found, the appliance checks the state of each of the corresponding GSLB services and load balances the GSLB services that are up.

3. If none of the perfect and partial matches are up, the appliance load balances all other available GSLB services.

   In this way, the appliance implements a type of site affinity for traffic that matches the DNS policy.

Example

Consider a GSLB configuration that consists of the following eight GSLB services:

- Asia.IN
- Asia.JPN
- Asia.HK
- Europe.UK
- Europe.RU
- Europe.EG
- Africa.SD
- Africa.ZMB

Further consider the following DNS action and policy configuration:

```
add dns action prefLoc11 GslbPrefLoc -preferredLocList "Asia.HK" "Europe.UK"

add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION("*.ZMB.*.*")" prefLoc11
<!--NeedCopy-->
```

When the appliance receives a request from the location
.ZMB..*, the preferred locations are evaluated as follows:

1. The appliance attempts to find a GSLB service whose location is a perfect match for Asia.HK, which is the preferred location that has the highest priority. It finds that the GSLB service at Asia.HK is a perfect match. If the GSLB service is up, it sends the client the IP address of the GSLB service.
2. If the GSLB service at Asia.HK is down, the appliance attempts to find a perfect match for the second preferred location, Europe.UK. It finds that the GSLB service at Europe.UK is a perfect match. If the GSLB service is up, it sends the client the IP address of the service.
3. If the GSLB service at Europe.UK is down, it returns to the preferred location that has the highest priority, Asia.HK, and looks for partial matches. For Asia.HK, it finds that Asia.IN and Asia.JPN are partial matches. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
4. If all partial matches for Asia.HK are down, the appliance looks for partial matches for Europe.UK. It finds that Europe.RU and Europe.EG are partial matches for the preferred location. If only one of the corresponding GSLB services is up, it sends the client the IP address of the service. If both locations are up, it load balances the two services.
5. If all partial matches for Europe.UK are down, the appliance load balances all other available GSLB services. In the current example, the appliance load balances Africa.SD and Africa.ZMB because the remaining six GSLB services have been found to be down.

Monitoring GSLB Services

September 21, 2020

When you bind a remote service to a GSLB virtual server, the GSLB sites exchange metric information, including network metric Information, which is the round-trip-time and persistence Information.

If a metric exchange connection is momentarily lost between any of the participating sites, the remote site is marked as DOWN and load balancing is performed on the remaining sites that are UP. When metric exchange for a site is DOWN, the remote services belonging to the site are marked DOWN as well.

The NetScaler appliance periodically evaluates the state of the remote GSLB services by using either MEP or monitors that are explicitly bound to the remote services. Binding explicit monitors to local services is not required, because the state of the local GSLB service is updated by default using the MEP. However, you can bind explicit monitors to a remote service. When monitors are explicitly bound, the state of the remote service is not controlled by the metric exchange.

By default, when you bind a monitor to a remote GSLB service, the NetScaler appliance uses the state of the service reported by the monitor. However, you can configure the NetScaler appliance to use monitors to evaluate services in the following situations:

- Always use monitors (default setting).
- Use monitors when MEP is DOWN.
- Use monitors when remote services and MEP are DOWN.

The second and third of the above settings enable the appliance to stop monitoring when MEP is UP. For example, in a hierarchical GSLB setup, a GSLB site provides the MEP information about its child sites to its parent site. Such an intermediate site may evaluate the state of the child site as DOWN because of network issues, though the actual state of the site is UP. In this case, you can bind monitors to the services of the parent site and disable MEP to determine the actual state of the remote service. This option enables you to control the manner in which the states of the remote services are determined.

To use monitors, first create them, and then bind them to GSLB services.

Configure monitor trigger

You can configure a GSLB site to always use monitors (the default), use monitors when MEP is down, or use monitors when both the remote service and MEP are down. In the latter two cases, the NetScaler appliance stops monitoring when MEP returns to the UP state.

To configure monitor triggering by using the command line interface

At the command prompt, type:

```
set gslb site <siteName> - triggerMonitor (ALWAYS | MEPDOWN |
    MEPDOWN_SVCDOWN)
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-North-America - triggerMonitor Always
<!--NeedCopy-->
```

To configure monitor triggering by using the configuration utility

1. Navigate to **Traffic Management** > **GSLB** > **Sites**, and double-click the site.
2. In the **Trigger Monitors** drop-down list, select an option for when to trigger monitoring.

Add or remove monitors

To add a monitor, you specify the type and the port. You cannot remove a monitor that is bound to a service. You must first unbind the monitor from the service.

To add a monitor by using the command line interface

At the command prompt, type the following commands to create a monitor and verify the configuration:

```
add lb monitor <monitorName> -type <monitorType> -destPort <portNumber>

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
<!--NeedCopy-->
```

To remove a monitor by using the command line interface

At the command prompt, type:

```
rm lb monitor <monitorName>
<!--NeedCopy-->
```

To add a monitor by using the configuration utility

Navigate to
Traffic Management > Load Balancing > Monitors, and add or delete a monitor.

Bind monitors to a GSLB service

Once you create monitors, you must bind them to GSLB services. When binding monitors to the services, you can specify a weight for the monitor. After binding one or more weighted monitors, you can configure a monitor threshold for the service. This threshold takes the service down if the sum of the bound monitor weights falls below the threshold value.

Note: In the configuration utility, you can set both the weight and the monitoring threshold at the same time that you bind the monitor. When using the command line, you must issue a separate command to set the service's monitoring threshold.

To bind the monitor to the GSLB service by using the command line interface

At the command prompt, type:

```
bind monitor <name> <serviceName> [ -state (Enabled | Disabled) ] -weight <positiveInteger>
<!--NeedCopy-->
```

Example:

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
<!--NeedCopy-->
```

To set the monitoring threshold for a GSLB service by using the command line interface

At the command prompt, type:

```
set gslb service <ServiceName> -monThreshold <PositiveInteger>
<!--NeedCopy-->
```

Example:

```
set gslb service service-GSLB-1 -monThreshold 9
<!--NeedCopy-->
```

To bind the monitor to the GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor** section and bind the monitor to the GSLB service.

To set the monitoring threshold for a GSLB service by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services.
2. Click the **Monitor Threshold** section and enter a threshold value.

Monitoring GSLB Sites

September 21, 2020

The NetScaler appliance uses MEP or monitors to determine the state of the GSLB sites. You can configure a GSLB site to always use monitors (the default), use monitors when MEP is down, or use monitors when both the remote service and MEP are down. In the latter two cases, the NetScaler appliance stops monitoring when MEP returns to the UP state.

To configure monitor triggering by using the command line interface

At the command prompt, type:

```
set gslb site <siteName>  - triggerMonitor (ALWAYS | MEPDOWN | MEPDOWN\
    _SVCDOWN)
<!--NeedCopy-->
```

Example:

```
set gslb site Site-GSLB-North-America  - triggerMonitor Always
<!--NeedCopy-->
```

To configure monitor triggering by using the configuration utility

1. Navigate to **Traffic Management**> **GSLB** > **Sites** and double-click the site.
2. In the **Trigger Monitors** drop-down list, select an option for when to trigger monitoring.

Protecting the GSLB Setup Against Failure

June 7, 2021

You can protect your GSLB setup against failure of a GSLB site or a GSLB virtual server by configuring a backup GSLB virtual server, configuring the NetScaler appliance to respond with multiple IP addresses, or configuring a Backup IP address for a GSLB domain. You can also divert excess traffic to a backup virtual server by using spillover.

Configuring a Backup GSLB Virtual Server

Configuring a backup entity for a GSLB virtual server ensures that DNS traffic to a site is not interrupted if the GSLB virtual server goes down. The backup entity can be another GSLB virtual server, or it can be a backup IP address. With a backup entity configured, if the primary GSLB virtual server goes down, the backup entity handles DNS requests. To specify what should happen when the primary GSLB virtual server comes back up again, you can configure the backup entity to continue handling traffic until you manually enable the primary virtual server to take over (using the disablePrimaryOnDown option).

Note: You can configure a single backup entity as a backup for multiple GSLB virtual server.

To configure a backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server as a backup virtual server and verify the configuration:

```
set gslb vserver <name> -backupVServer <name> [-disablePrimaryOnDown (
    ENABLED | DISABLED)]

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -
    disablePrimaryOnDown ENABLED

show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

To set GSLB virtual server as a backup virtual server by using the configuration utility

1. Navigate to **Traffic Management > GSLB > Virtual Servers**, and double-click the GSLB virtual server.
2. Click the **Backup Virtual Server** section and select the backup virtual server.

Configuring a GSLB Setup to Respond with Multiple IP Addresses

A typical DNS response contains the IP address of the best performing GSLB service. However, if you enable multiple IP response (MIR), the NetScaler appliance sends the best GSLB service as the first record in the response and adds the remaining active services as additional records. If MIR is disabled (the default), the NetScaler appliance sends the best service as the only record in the response.

To configure a GSLB virtual server for multiple IP responses by using the command line interface

At the command prompt, type the following commands to configure a GSLB virtual server for multiple IP responses and verify the configuration:

```
set gslb vserver<name> -MIR (ENABLED | DISABLED)

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED

show gslb vserver <vserverName>
<!--NeedCopy-->
```

To set a GSLB virtual server for multiple IP responses by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the Advanced tab, under When this VServer is "UP," select the Send all "active" service IP in response (MIR) check box, and click OK.

Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN

A DNS response can contain either the IP address of the requested domain or an answer stating that the IP address for the domain is not known by the DNS server, in which case the query is forwarded to another name server. These are the only possible responses to a DNS query.

When a GSLB virtual server is disabled or in a DOWN state, the response to a DNS query for the GSLB domain bound to that virtual server contains the IP addresses of all the services bound to the virtual server. However, you can configure the GSLB virtual server to in this case send an empty down response (EDR). When this option is set, a DNS response from a GSLB virtual server that is in a DOWN state does not contain IP address records, but the response code is successful. This prevents clients from attempting to connect to GSLB sites that are down.

Note: You must configure this setting for each virtual server to which you want it to apply.

To configure a GSLB virtual server for empty down responses by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -EDR (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -EDR ENABLED
<!--NeedCopy-->
```

To set a GSLB virtual server for empty down responses by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for which you want to configure a backup virtual server (for example, vserver-GSLB-1).
2. On the Advanced tab, under When this VServer is "Down," select the Do not send any service's IP address in response (EDR) check box.
3. Click OK.

Configuring a Backup IP Address for a GSLB Domain

You can configure a backup site for your GSLB configuration. With this configuration in place, if all of the primary sites go DOWN, the IP address of the backup site is provided in the DNS response.

Typically, if a GSLB virtual server is active, that virtual server sends a DNS response with one of the active site IP addresses as selected by the configured GSLB method. If all the configured primary sites in the GSLB virtual server are inactive (in the DOWN state), the authoritative domain name system (ADNS) server or DNS server sends a DNS response with the backup site's IP address.

Note: When a backup IP address is sent, persistence is not honored.

To set a backup IP address for a domain by using the command line interface

At the command prompt, type the following commands to set a backup IP address and verify the configuration:

```
set gslb vserver <name> -domainName <string> -backupIP <IPAddress>

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP
    10.102.29.66

show gslb vserver vserver-GSLB-1
<!--NeedCopy-->
```

To set a backup IP address for a domain by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server to which you want to bind the backup domain (for example, vserver-GSLB-1).
2. Click the **Domains** section, configure the GSLB domain and specify the IP address of the backup domain in the **Backup IP** field.

Diverting Excess Traffic to a Backup Virtual Server

Once the number of connections to a primary GSLB virtual server exceeds the configured threshold value, you can use the spillover option to divert new connections to a backup GSLB virtual server. This threshold value can be calculated dynamically or set manually. Once the number of connections to the primary virtual server drops below the threshold, the primary GSLB virtual server resumes serving client requests.

You can configure persistence with spillover. When persistence is configured, new clients are diverted to the backup virtual server if that client is not already connected to a primary virtual server. When persistence is configured, connections that were diverted to the backup virtual server are not moved back to the primary virtual server after the number of connections to the primary virtual server drops below the threshold. Instead, the backup virtual server continues to process those connections until they are terminated by the user. Meanwhile, the primary virtual server accepts new clients.

The threshold can be measured by the number of connections, bandwidth, and health of the services.

If the backup virtual server reaches the configured threshold and is unable to take any additional load, the primary virtual server diverts all requests to the designated redirect URL. If a redirect URL is not configured on the primary virtual server, subsequent requests are dropped.

The spillover feature prevents the remote backup GSLB service (backup GSLB site) from getting flooded with client requests when the primary GSLB virtual server fails. This occurs when a monitor is bound to a remote GSLB service, and the service experiences a failure that causes its state to go DOWN. The monitor continues to keep the state of the remote GSLB service UP, however, because of the spillover feature.

As part of the resolution to this problem, two states are maintained for a GSLB service, the primary state and effective state. The primary state is the state of the primary virtual server and the effective state is the cumulative state of the virtual servers (primary and backup chain). The effective state is set to UP if any of the virtual servers in the chain of virtual servers is UP. A flag that indicates that the primary VIP has reached the threshold is also provided. The threshold can be measured by either the number of connections or the bandwidth.

A service is considered for GSLB only if its primary state is UP. Traffic is directed to the backup GSLB service only when all the primary virtual servers are DOWN. Typically, such deployments will have only one backup GSLB service.

Adding primary and effective states to a GSLB service has the following effects:

- When source IP persistence is configured, the local DNS is directed to the previously selected site only if the primary virtual server on the selected site is UP and below threshold. Persistence can be ignored in the round robin mode.
- If cookie-based persistence is configured, client requests are redirected only when the primary virtual server on the selected site is UP.
- If the primary virtual server has reached its saturation and the backup VIP(s) is absent or down, the effective state is set to DOWN.
- If external monitors are bound to an HTTP-HTTPS virtual server, the monitor decides the primary state.
- If there is no backup virtual server to the primary virtual server and the primary virtual server has reached its threshold, the effective state is set to DOWN.

To configure a backup GSLB virtual server by using the command line interface

At the command prompt, type the following commands to configure a backup GSLB virtual server and verify the configuration:

```
set gslb vserver <name> -soMethod <method> -soThreshold <threshold> -soPersistence ( **ENABLED** | **DISABLED** ) -soPersistenceTimeout <timeout>

show gslb vserver <name>
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000 -soPersistence ENABLED -soPersistenceTimeout 2

show gslb vserver Vserver-GSLB-1
<!--NeedCopy-->
```

To configure a backup GSLB virtual server by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server that you want to configure as a backup (for example, Vserver-LB-1).
2. Click the **SpillOver** section and set the following parameters:
    - Method— soMethod
    - Threshold— soThreshold
    - Persistence Time-out (min) — soPersistenceTimeout
3. Select the Persistence option and click **OK**.

Managing Client Connections

September 21, 2020

Aug 07, 2017

To facilitate management of client connections, you can enable delayed cleanup of connections to the virtual server. You can then manage local DNS traffic by configuring DNS policies.

Enabling Delayed Cleanup of Virtual Server Connections

The state of a virtual server depends on the states of the services bound to it, and the state of each service depends on the monitors bound to it. If a server is slow or down, the monitoring probes time out and the service that represents the server is marked as DOWN. A virtual server is marked as DOWN only when all services bound to it are marked as DOWN. You can configure services and virtual servers to either terminate all connections when they go down, or allow the connections to go through. The latter setting is for situations in which a service is marked as DOWN because of a slow server.

When you configure the down state flush option, the NetScaler appliance performs a delayed cleanup of connections to a GSLB service that is down.

To enable delayed cleanup of virtual server connections by using the command line interface

At the command prompt, type the following commands to configure delayed connection cleanup and verify the configuration:

```
set gslb service <name> -downStateFlush (ENABLED | DISABLED)
show gslb service <name>
<!--NeedCopy-->
```

Example:

```
set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done

show gslb service Service-GSLB-1
Done
<!--NeedCopy-->
```

To enable delayed cleanup of virtual server connections by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services and double-click the service.
2. Click the **Other Settings** section and select the **Down State Flush** option.

Managing Local DNS Traffic by Using DNS Policies

You can use DNS policies to implement site affinity by directing traffic from the IP address of a local DNS resolver or network to a predefined target GSLB site. This is configured by creating DNS policies with DNS expressions and binding the policies globally on the NetScaler appliance.

DNS Expressions

The NetScaler appliance provides certain predefined DNS expressions that can be used for configuring actions specific to a domain. Such actions can, for example, drop certain requests, select a specific view for a specific domain, or redirect certain requests to a specific location.

These DNS expressions (also called *rules*) are combined to create DNS policies that are then bound globally on the NetScaler appliance.

Following is the list of predefined DNS qualifiers available on the NetScaler appliance:

- CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
- CLIENT.UDP.DNS.IS_AREC
- CLIENT.UDP.DNS.IS_AAAAREC
- CLIENT.UDP.DNS.IS_SRVREC
- CLIENT.UDP.DNS.IS_MXREC
- CLIENT.UDP.DNS.IS_SOAREC
- CLIENT.UDP.DNS.IS_PTRREC
- CLIENT.UDP.DNS.IS_CNAME
- CLIENT.UDP.DNS.IS_NSREC
- CLIENT.UDP.DNS.IS_ANYREC

The CLIENT.UDP.DNS.DOMAIN DNS expression can be used with string expressions. If you are using domain names as part of the expression, they must end with a period (.). For example, CLIENT.UDP.DNS.DOMAIN.ENDSWITH("abc.com.")

To create an expression by using the configuration utility

1. Click the icon next to the Expression text box. Click Add. (Leave the Flow Type and Protocol drop-down list boxes empty.) Follow these steps to create a rule.
2. In the Qualifier box, select a qualifier (for example, LOCATION).
3. In the Operator box, select an operator (for example, ==).
4. In the Value box, type a value (for example, Asia, Japan….).
5. Click OK. Click Create and click Close. The rule is created.
6. Click OK.

Configuring DNS Actions

A DNS policy includes the name of a DNS action to be performed when the policy rule evaluates to TRUE. A DNS action can do one of the following:

- Send the client an IP address for which you have configured a DNS view. For more information about DNS views, see Adding DNS Views.
- Send the client the IP address of a GSLB service after referring to a list of preferred locations that overrides static proximity behavior. For more information about preferred locations, see Overriding Static Proximity Behavior by Configuring Preferred Locations.
- Send the client a specific IP address as determined by the evaluation of the DNS query or response (DNS response rewrite).
- Forward a request to the name server without performing a lookup in the appliance's DNS cache.
- Drop a request.

You cannot create a DNS action for dropping a DNS request or for bypassing the DNS cache on the appliance. If you want to drop a DNS request, use the built-in action, dns_default_act_Drop. If you want to bypass the DNS cache, use the built-in action, dns_default_act_Cachebypass. Both actions are available along with custom actions in the Create DNS Policy and the Configure DNS Policy dialog boxes. These built-in actions cannot be modified or removed.

To configure a DNS action by using the command line interface

At the command prompt, type the following commands to configure a DNS action and verify the configuration:

```
add dns action <actionName> <actionType> (-IPAddress <ip_addr | ipv6_addr> ... | -viewName <string> | -preferredLocList <string> ...) [-TTL <secs>]

show dns action [<actionName>]
<!--NeedCopy-->
```

Examples:

Example 1: Configuring DNS Response Rewrite. The following DNS action sends the client a preconfigured IP address when the policy to which the action is bound evaluates to true:

```
add dns action dns_act_response_rewrite Rewrite_Response -IPAddress 192.0.2.20 192.0.2.56 198.51.100.10
 Done

show dns action dns_act_response_rewrite
1)   ActionName:   dns_act_response_rewrite ActionType:   Rewrite_Response    TTL:   3600                 IPAddress:   192.0.2.20 192.0.2.56 198.51.100.10
 Done
<!--NeedCopy-->
```

Example 2: Configuring a DNS-View Based Response. The following DNS action sends the client an IP address for which you have configured a DNS view:

```
add dns action send_ip_from_view_internal_ip ViewName -viewName view_internal_ip
Done

show dns action send_ip_from_view_internal_ip
1)   ActionName:   send_ip_from_view_internal_ip ActionType:   ViewName       ViewName:   view_internal_ip
 Done
<!--NeedCopy-->
```

Example 3: Configuring a Response Based on a Preferred Location List. The following DNS action sends the client the IP address that corresponds to the preferred location that it selects from the specified list of locations:

```
add dns action send_preferred_location GslbPrefLoc -preferredLocList NA.tx.ns1.*.*.* NA.tx.ns2.*.*.* NA.tx.ns3.*.*.*
Done

show dns action  send_preferred_location
```

```
1)  ActionName:  send_preferred_location ActionType:  GslbPrefLoc PreferredLocList:  "NA.tx.ns1.*.*.*" "NA.tx.ns2.*.*.*" "NA.tx.ns3.*.*.*"
 Done
<!--NeedCopy-->
```

To configure a DNS action by using the NetScaler configuration utility

1. Navigate to Traffic Management > DNS > Actions, create or edit a DNS action.
2. In the Create DNS Action or Configure DNS Action dialog box, set the following parameters:
    - Action Name (cannot be changed for an existing DNS action)
    - Type (cannot be changed for an existing DNS action)
      To set the
      Type parameter, do one of the following:
        - To create a DNS action that is associated with a DNS view, select View Name. Then, from the View Name list, select the DNS view that you want to use in the action.
        - To create a DNS action with a preferred location list, select Preferred Location List. In Preferred Location, enter a location, and then click Add. Add as many DNS locations as you want.
        - To configure a DNS action for rewriting a DNS response on the basis of policy evaluation, select Rewrite Response. In IP Address, enter an IP address, and then click Add. Add as many IP addresses as you want.
    - TTL (applicable only to the Rewrite Response action type)

Configuring DNS Policies

DNS policies operate on a location database that uses static and custom IP addresses. The attributes of the incoming local DNS request are defined as part of an expression, and the target site is defined as part of a DNS policy. While defining actions and expressions, you can use a pair of single quotation marks (‘’) as a wildcard qualifier to specify more than one location. When a DNS policy is configured and a GSLB request is received, the custom IP address database is first queried for an entry that defines the location attributes for the source:

- When a DNS query comes from an LDNS, the characteristics of the LDNS are evaluated against the configured policies. If they match, an appropriate action (site affinity) is executed. If the LDNS characteristics match more than one site, the request is load balanced between the sites that match the LDNS characteristics.
- If the entry is not found in the custom database, the static IP address database is queried for an entry, and if there is a match, the above policy evaluation is repeated.
- If the entry is not found in either the custom or static databases, the best site is selected and sent in the DNS response on the basis of the configured load balancing method.

The following restrictions apply to DNS policies created on the NetScaler appliance.

- A maximum of 64 policies are supported.
- DNS policies are global to the NetScaler and cannot be applied to a specific virtual server or domain.
- Domain or virtual server specific binding of policy is not supported.

You can use DNS policies to direct clients that match a certain IP address range to a specific site. For example, if you have a GSLB setup with multiple GSLB sites that are separated geographically, you can direct all clients whose IP address is within a specific range to a particular data center.

Both TCP-based and UDP-based DNS traffic can be evaluated. Policy expressions are available for UDP-based DNS traffic on the server and for both UDP-based DNS traffic and TCP-based DNS traffic on the client side. Additionally, you can configure expressions to evaluate queries and responses that involve only the following DNS question types (or QTYPE values):

- A
- AAAA
- NS
- SRV
- PTR
- CNAME
- SOA
- MX
- ANY

The following response codes (RCODE values) are also supported:

- NOERROR - No error
- FORMERR - Format error
- SERVFAIL - Server failure
- NXDOMAIN - Non-existent domain
- NOTIMP - Query type not implemented
- REFUSED - Query refused

You can configure expressions to evaluate DNS traffic. A DNS expression begins with the DNS.REQ or DNS.RES prefixes. Functions are available for evaluating the queried domain, the query type, and the carrier protocol. For more information about DNS expressions, see "Expressions for Evaluating a DNS Message and Identifying Its Carrier Protocol" in "Policy Configuration and Reference".

To add a DNS policy by using the command line interface

At the command prompt, type the following commands to create a DNS policy and verify the configuration:

```
add dns policy <name> <rule> <actionName>
show dns policy <name>
<!--NeedCopy-->
```

Example:

```
> add dns policy policy-GSLB-1 'CLIENT.UDP.DNS.DOMAIN.EQ("domainname")'
    my_dns_action
 Done
> show dns policy policy-GSLB-1
    Name: policy-GSLB-1
    Rule: CLIENT.UDP.DNS.DOMAIN.EQ("domainname")
    Action Name: my_dns_action
    Hits: 0
    Undef Hits: 0

 Done
<!--NeedCopy-->
```

To remove a configured DNS policy by using the command line interface

At the command prompt, type:

```
rm dns policy <name>
<!--NeedCopy-->
```

To configure a DNS policy by using the NetScaler configuration utility

1. Navigate to Traffic Management > DNS > Policies and create a DNS policy.
2. In the Create DNS Policy or Configure DNS Policy dialog box, set the following parameters:
    - Policy Name (cannot be changed for an existing policy)
    - Action
    - Expression

      To specify an expression, do the following:

        a) Click Add, and then, in the drop-down box that appears, select the expression element with which you want to begin the expression. A second list appears. The list contains a set of expression elements that you can use immediately after the firs expression element.

        b) In the second list, select the expression element that you want, and then enter a period.

   c) After each selection, if you enter a period, the next set of valid expression elements appear in a list. Select expression elements and fill in arguments to functions until you have the expression you want.
3. Click Create or OK, and then click Close.

Binding DNS Policies

DNS policies are bound globally on the NetScaler appliance and are available for all configured GSLB virtual servers. Even though DNS policies are globally bound, policy execution can be limited to a specific GSLB virtual server by specifying the domain in the expression.

Note: Even though the bind dns global command accepts
REQ_OVERRIDE and
RES_OVERRIDE as valid bind points, those bind points are redundant, because DNS policies can be bound only globally. Bind your DNS policies only to the
REQ_DEFAULT and
RES_DEFAULT bind points.

To bind a DNS policy globally by using the command line interface

At the command prompt, type the following commands to bind a DNS policy globally and verify the configuration:

At the command prompt, type the following commands to bind a DNS policy globally and verify the configuration:

```
bind dns global <policyName> <priority> [-gotoPriorityExpression <string>] [-type <type>]
show dns global -type <type>
<!--NeedCopy-->
```

Example:

```
bind dns global policy-GSLB-1 10 -gotoPriorityExpression END
 Done
show dns global -type REQ_DEFAULT
1)  Policy Name: policy-GSLB-1
    Priority: 10
    GotoPriorityExpression: END
 Done
<!--NeedCopy-->
```

To bind a DNS policy globally by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind DNS Policy(s) to Global dialog box, click Insert Policy.
4. In the Policy Name column, select, from the list, the policy that you want to bind. Alternatively, in the list, click New Policy, and then create a DNS policy by setting parameters in the Create DNS Policy dialog box.
5. To modify a policy that is already bound globally, click the name of the policy, and then click Modify Policy. Then, in the Configure DNS Policy dialog box, modify the policy, and then click OK.
6. To unbind a policy, click the name of the policy, and then click Unbind Policy.
7. To modify the priority assigned to a policy, double-click the priority value, and then enter a new value.
8. To regenerate assigned priorities, click Regenerate Priorities. The priority values are modified to begin at 100, with increments of 10, without affecting the order of evaluation.
9. Click OK.

To view the global bindings of a DNS policy by using the command line interface

At the command prompt, type:

```
show dns global
<!--NeedCopy-->
```

To view the global bindings of a DNS policy by using the configuration utility

1. Navigate to Traffic Management > DNS > Policies.
2. In the details pane, click Global Bindings. The global bindings of all DNS policies appear in this dialog box.

Adding DNS Views

You can configure DNS views to identify various types of clients and provide an appropriate IP address to a group of clients who query for the same GSLB domain. DNS views are configured by using DNS policies that select the IP addresses sent back to the client.

For example, if you have configured GSLB for your company's domain and have the server hosted in your company's network, clients querying for the domain from within your company's internal network can be provided with the server's internal IP address instead of the public IP address. Clients that query DNS for the domain from the Internet, on the other hand, can be provided the domain's public IP address.

To add a DNS view, you assign it a name of up to 31 characters. The leading character must be a number or letter. The following characters are also allowed: @ _ - . (period) : (colon) # and space ( ). After adding the view, you configure a policy to associate it with clients and a part of the network, and you bind the policy globally. To configure and bind a DNS policy, see Managing Local DNS Traffic by Using DNS Policies.

To add a DNS view by using the command line interface

At the command prompt, type the following commands to create a DNS view and verify the configuration:

```
add dns view <viewName>
show dns view <viewName>
<!--NeedCopy-->
```

Example:

```
add dns view PrivateSubnet
show dns view PrivateSubnet
<!--NeedCopy-->
```

To remove a DNS view by using the command line interface

At the command prompt, type:

```
rm dns view <viewName>
<!--NeedCopy-->
```

To add a DNS view by using the configuration utility

Navigate to Traffic Management > DNS > Views and add a DNS view.

For details on how to create a DNS policy and how to bind DNS policies globally, see Managing Local DNS Traffic by Using DNS Policies.

Configuring GSLB for Disaster Recovery

September 21, 2020

Disaster recovery capability is critical, because downtime is costly. A NetScaler appliance configured for GSLB forwards traffic to the least-loaded or the best-performing data center. This configuration, referred to as an active-active setup, not only improves performance, but also provides immediate disaster recovery by routing traffic to other data centers if a data center that is part of the setup goes down. Alternatively, you can configure an active-standby GSLB setup for disaster recovery only.

Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup

A conventional disaster recovery setup includes an active data center and a standby data center. The standby data center is a remote site. When a failover occurs as a result of a disaster event that causes the primary active data center to be inactive, the standby data center becomes operational.

Configuring disaster recovery in an active-standby data-center setup consists of the following tasks.

- Create the active data center.
  - Add a local GSLB site.
  - Add a GSLB vserver, which represents the active data center.
  - Bind the domain to the GSLB virtual server.
  - Add gslb services and bind the services to active GSLB virtual server.
- Create the standby data center.
  - Add a remote gslb site.
  - Add a gslb vserver, which represents standby data center.
  - Add gslb services which represents standby data center and bind the services to the standby gslb vserver.
  - Designate the standby data center by configuring the standby GSLB virtual server as the backup virtual server for the active GSLB virtual server.

Once you have configured the primary data center, replicate the configuration for the backup data center and designate it as the standby GSLB site by designating a GSLB virtual server at that site as the backup virtual server.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

To designate the standby GSLB site by using the command line interface

At both the active site and the remote site, at the command prompt, type:

```
set gslb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

Example:

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
<!--NeedCopy-->
```

To configure the standby site by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the GSLB virtual server for the primary site.
2. Click the **Backup Virtual Server** section and select a backup virtual server.

By default, once the primary virtual server becomes active, it starts receiving traffic. However, if you want the traffic to be directed to the backup virtual server even after the primary virtual server becomes active, use the 'disable primary on down' option.

Configuring for Disaster Recovery in an Active-Active Data Center Setup

An active-active GSLB deployment, in which both GSLB sites are active, removes any risk that may arise in having a standby data center. With such a setup, web or application content can be mirrored in geographically separate locations. This ensures that data is consistently available at each distributed data center.

To configure GSLB for disaster recovery in an active-active data center set up, you must first configure the basic GSLB setup on the first data center and then configure all other data centers.

First create at least two GSLB sites. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server in the local site. Finally, at the local site, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Once you have configured the first data center, replicate the configuration for other data centers part of the setup.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Configuring for Disaster Recovery with Weighted Round Robin

When you configure GSLB to use the weighted round robin method, weights are added to the GSLB services and the configured percentage of incoming traffic is sent to each GSLB site. For example, you can configure your GSLB setup to forward 80 percent of the traffic to one site and 20 percent of the traffic to another. After you do this, the NetScaler appliance will send four requests to the first site for each request that it sends to the second.

To set up the weighted round robin method, first create two GSLB sites, local and remote. Next, for the local site create a GSLB virtual server and GSLB services, and bind the services to the virtual server. Configure the GSLB method as round robin. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

Each service that represents a physical server in the network has weights associated with it. Therefore the GSLB service is assigned a dynamic weight that is the sum of weights of all services bound to it. Traffic is then split between the GSLB services based on the ratio of the dynamic weight of the particular service to the total weight. You can also configure individual weights for each GSLB service instead of the dynamic weight.

If the services do not have weights associated with them, you can configure the GSLB virtual server to use the number of services bound to it to calculate the weight dynamically.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you configure a basic GSLB setup, you must configure the weighted round robin method such that the traffic is split between the configured GSLB sites according to the weights configured for the individual services.

To configure a virtual server to assign weights to services by using the command line interface

At the command prompt, type one of the following commands, depending upon whether you want to create a new load balancing virtual server or configure an existing one:

```
add lb vserver <name>@ -weight <WeightValue> <ServiceName>

set lb vserver <name>@ -weight <WeightValue> <ServiceName>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1

set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
<!--NeedCopy-->
```

To set dynamic weight by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
<!--NeedCopy-->
```

Example:

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
<!--NeedCopy-->
```

To add weights to the GSLB services by using the command line interface

At the command prompt, type:

set gslb vserver -serviceName GSLBServiceName -weight WeightValue

Example:

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
<!--NeedCopy-->
```

To configure a virtual server to assign weights to services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and double-click the virtual server (for example, Vserver-LB-1).
2. Click the **Services** section and set the weight of a service.

To add weights to the GSLB services by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server (for example, vserver-GSLB-1)
2. Click the **Services** section and set the weight of the service in the **Weight** field.

To set dynamic weight by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server (for example, vserver-GSLB-1).
2. Click the **Method** section and, from the **Dynamic Weight** drop-down list select **SERVICEWEIGHT**.

Configuring for Disaster Recovery with Data Center Persistence

Data center persistence is required for web applications that require maintaining a connection with the same server instead of having the requests load balanced. For example, in an e-commerce portal, maintaining a connection between the client and the same server is critical. For such applications, HTTP redirect persistence can be configured in an active-active setup.

To configure GSLB for disaster recovery with data center persistence, you must first configure the basic GSLB set up and then configure HTTP redirect persistence.

First create two GSLB sites, local and remote. Next, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Next, create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Next, create a load balancing virtual server with the same virtual server IP address as the GSLB service. Finally, duplicate the previous steps for the remote configuration, or configure the NetScaler appliance to autosynchronize your GSLB configuration.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure HTTP redirect precedence to enable data center persistence.

To configure HTTP redirect by using the command line interface

At the command prompt, type the following commands to configure HTTP redirect and verify the configuration:

```
set gslb service <serviceName> -sitePersistence <sitePersistence> -sitePrefix <string>

show gslb service <serviceName>
<!--NeedCopy-->
```

Example:

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1

show gslb service Service-GSLB-1
<!--NeedCopy-->
```

To configure HTTP redirect by using the configuration utility

1. Navigate to Traffic Management > GSLB > Services and double-click the GSLB service to be configured.
2. Click the **Site Persistence** section, select the **HTTPRedirect** option, and in the **Site Prefix** text box, enter the site prefix (for example, vserver-GSLB-1).

> **Note**
>
> When site persistence is not configured and if a load balancing virtual server that is configured as a local GSLB service is DOWN, the HTTP requests are redirected to other healthy GSLB sites using a 302 redirect.

Configuring GSLB for Proximity

September 21, 2020

When you configure GSLB for proximity, client requests are forwarded to the closest data center. The main benefit of the proximity-based GSLB method is faster response times resulting from the selection of the closest available data center. Such a deployment is critical for applications that require fast access to large volumes of data.

You can configure GSLB for proximity based on the round trip time (RTT), static proximity, or a combination of the two.

Configuring Dynamic Method (RTT)

Dynamic round trip time (RTT) is a measure of time or delay in the network between the client's local DNS server and a data resource. To measure dynamic RTT, the NetScaler appliance probes the client's local DNS server and gathers RTT metric information. The NetScaler then uses this metric to make its load balancing decision. Global server load balancing monitors the real-time status of the network and dynamically directs the client request to the data center with the lowest RTT value

To configure GSLB for proximity with dynamic method, you must first configure the basic GSLB set up and then configure dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the dynamic RTT method.

For details on how to configure the GSLB virtual server to use the dynamic RTT method for load balancing, see Configuring Dynamic RTT.

Configuring Static Proximity

The static proximity method for GSLB uses an IP address-based static proximity database to determine the proximity between the client's local DNS server and the GSLB sites. The NetScaler appliance responds with the IP address of a site that best matches the proximity criteria.

If two or more GSLB sites at different geographic locations serve the same content, the NetScaler appliance maintains a database of IP address ranges and uses the database for decisions about the GSLB sites to which to direct incoming client requests.

To configure GSLB for proximity with static proximity, you must first configure the basic GSLB set up and then configure static proximity.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure static proximity.

For details on how to configure the GSLB virtual server to use static proximity for load balancing, see Configuring Static Proximity.

Configuring Static Proximity and Dynamic RTT

You can configure the GSLB virtual server to use a combination of static proximity and dynamic RTT when you have some clients coming from an internal network like a branch office. You can configure GSLB such that the clients coming from the branch office or any other internal network are directed to a particular GSLB site that is geographically close to the client network. For all other requests, you can use dynamic RTT.

First create two GSLB sites, local and remote. Then, for the local site, create a GSLB virtual server and GSLB services and bind the services to the virtual server. Then create ADNS services and bind the domain for which you are configuring GSLB to the GSLB virtual server at the local site. Finally, create a load balancing virtual server with the same virtual server IP address as the GSLB service.

For details on how to configure a basic GSLB setup, see Configuring Global Server Load Balancing (GSLB).

Once you have configured a basic GSLB setup, configure the GSLB virtual server to use static proximity for all traffic originating from an internal network and then use dynamic RTT for all other traffic.

For details on how to configure static proximity, see Configuring Static Proximity and for details on how to configure dynamic RTT, see Configuring Dynamic RTT.

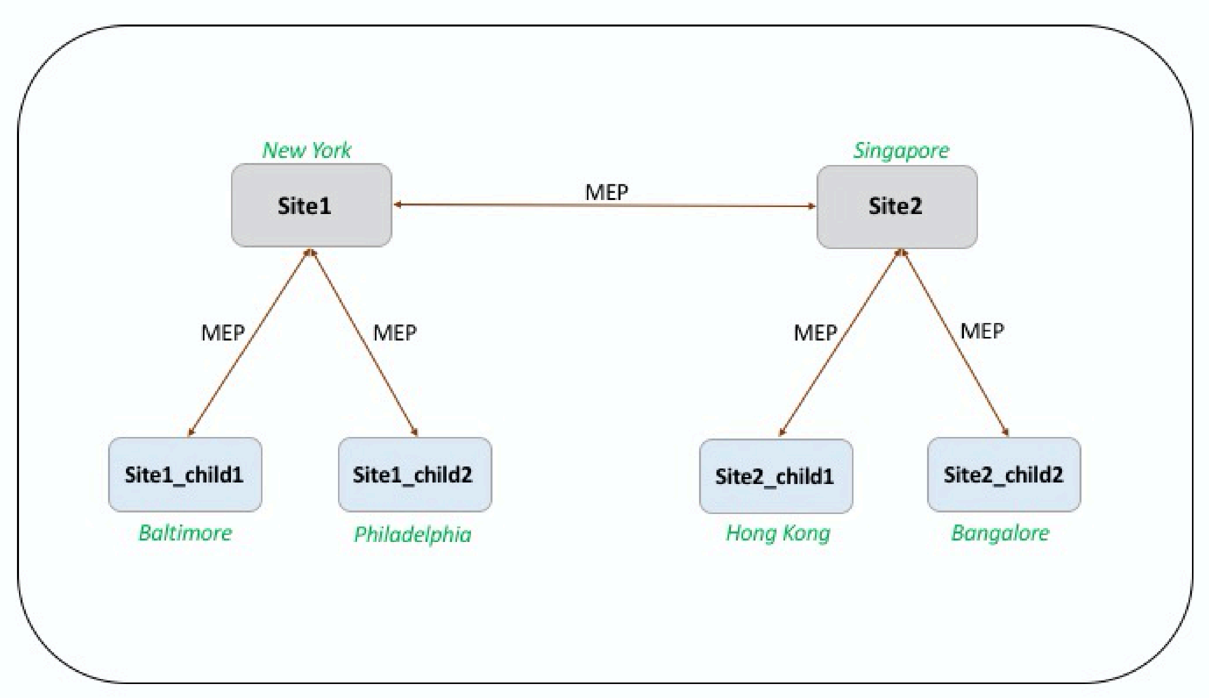
Example of a Complete Parent-Child Configuration Using the Metrics Exchange Protocol

September 21, 2020

Jun 19, 2018

Consider the following parent-child topology in which the GSLB sites are distributed globally.

- Site1 and Site2 are the parent sites.
- Site1_child1 and Site1_child2 are the child sites of Site1.
- Site2_child1 and Site2_child2 are the child sites of Site2.


New York
Site1
MEP
Singapore
Site2
MEP
MEP
MEP
MEP
Site1_child1
Baltimore
Site1_child2
Philadelphia
Site2_child1
Hong Kong
Site2_child2
Bangalore

The following commands illustrate the complete configuration of the parent-child topology.

site1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2

```

```
add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP 10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP 10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -appflowLog DISABLED

bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1

bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1

bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
<!--NeedCopy-->
```

site1_child1

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite site1
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.132 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site1_child2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

You can add the following commands for load balancing configuration:

add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp OFF -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.102.82.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2

```
add gslb site site1 10.102.82.164 -publicIP 10.102.82.164

add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site1_child1 10.102.82.131 -publicIP 10.102.82.131 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site1

```

```
add gslb site site1_child2 10.102.82.67 -publicIP 10.102.82.67 -nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite site1

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite site2

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite site2

add gslb service site1_child1_http_gsvc1 10.102.82.132 HTTP 80 -publicIP 10.102.82.132 -publicPort 80 -maxClient 0 -siteName site1_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site1_child2_http_gsvc1 10.102.82.68 HTTP 80 -publicIP 10.102.82.68 -publicPort 80 -maxClient 0 -siteName site1_child2 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child1_http_gsvc1 10.106.24.134 HTTP 80 -publicIP 10.106.24.134 -publicPort 80 -maxClient 0 -siteName site2_child1 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb service site2_child2_http_gsvc1 10.106.24.68 HTTP 80 -publicIP 10.106.24.68 -publicPort 80 -maxClient 0 -siteName site2_child2 -cltTimeout 180 -svrTimeout 360 -downStateFlush ENABLED

add gslb vserver gv1 HTTP -backupLBMethod ROUNDROBIN -tolerance 0 -appflowLog DISABLED

bind gslb vserver gv1 -serviceName site1_child1_http_gsvc1

bind gslb vserver gv1 -serviceName site1_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child2_http_gsvc1

bind gslb vserver gv1 -serviceName site2_child1_http_gsvc1

bind gslb vserver gv1 -domainName www.gslb.com -TTL 5
<!--NeedCopy-->
```

site2_child1

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child1 10.106.24.132 -publicIP 10.106.24.132 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.134 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

site2_child2

```
add gslb site site2 10.106.24.164 -publicIP 10.106.24.164

add gslb site site2_child2 10.106.24.67 -publicIP 10.106.24.67 -
    nwMetricExchange DISABLED -sessionExchange DISABLED -parentSite
    site2
<!--NeedCopy-->
```

You can add the following commands for load balancing configuration:

```
add service svc1 10.102.82.25 HTTP 80 -gslb NONE -maxClient 0 -maxReq 0
     -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -
    svrTimeout 360 -CKA NO -TCPB NO -CMP NO

add lb vserver lb1 HTTP 10.106.24.68 80 -persistenceType NONE -
    cltTimeout 180

bind lb vserver lb1 svc1
<!--NeedCopy-->
```

Configuring GSLB Service Selection Using Content Switching

September 21, 2020

In a typical GSLB deployment, you can prioritize the selection of a set of GSLB services bound to a GSLB virtual server, but you cannot do the following:

- Restrict the selection of a GSLB service from a subset of GSLB services bound to a GSLB virtual server for the given domain.
- Apply different load balancing methods on the different subsets of GSLB services in the deployment.
- Apply spillover policies on a subset of GSLB services, and you cannot have a backup for a subset of GSLB services.
- Configure a subset of GSLB services to serve different content. That is, you cannot content switch between servers in different GSLB sites. The GSLB configuration assumes that the servers contain the same content.
- Define a subset GSLB services with different priorities and specify an order in which the services in the subset are applied to a request.

You can now configure a content switching (CS) policy to customize the GSLB deployment. First configure a set of GSLB services and bind it to a GSLB virtual server. Then, configure a CS virtual server of target type GSLB, define a CS policy and action with the GSLB virtual server as target virtual server, and bind the CS policy to CS virtual server.

> **Important**
>
> - Only CS policies with DNS based expressions can be bound to a CS virtual server of target type GSLB.
> - If a GLSB service is bound to a CS virtual server through a GSLB virtual server, you cannot bind another GSLB virtual server bound with the same GSLB service to the CS virtual server.

Example:
Consider a GLSB deployment that includes two GSLB sites. At each site, four GSLB services (S-1, S-2, S-3, and S-4) are bound to GSLB virtual server VS-1. You can configure a content switching (CS) virtual server of target type GSLB and define a CS policy and action with VS-1 as the target virtual server, so that requests for content in English are served only by S-1 and S-2, and requests for content in the local language are served only by S-3 and S-4.

You could give S-1 priority by configuring a backup virtual server to VS-1 and binding S-2 to the backup virtual server. Client requests would then be served by S-1 unless the server it represents went down, in which case the requests would be served by S-2. If both S-1 and S-2 were down, clients would receive an empty response.

To configure GSLB Service Selection using Content Switching:

1. Configure GSLB. For instructions, see Configuring Global Server Load Balancing.
2. Configure a Content Switching (CS) virtual server of target type GSLB. For more information, see Creating Content Switching Virtual Servers.
3. Configure Content Switching (CS) policies. For more information, see Configuring Content Switching Policies.
4. Configure CS actions that designate a GSLB virtual server as the target virtual server. For more information, see Configuring a Content Switching Action.
5. Bind the CS policies to the CS virtual server. For more information, see Binding Policies to a Content Switching Virtual Server.
6. Bind the domain to the CS virtual server instead of the GSLB virtual server.

Sample Configuration

The following sample configuration sends requests from the client with IP address 5.5.5.5 to SERVICE_GSLB1 and SERVICE_GSLB2. SERVICE_GSLB1 has a higher priority than SERVICE_GSLB2, and SERVICE_GSLB2 serves the client requests only when SERVICE_GSLB1 is down. If both SERVICE_GSLB1 and SERVICE_GSLB2 are down, SERVICE_GSLB3 and service-GSLB4 are not considered, and a blank response is sent to the client.

```
>add cs vs CSVSERVER_GSLB http  – targettype GSLB
Done
>add gslb vs VSERVER_GSLB1 http
Done
>add gslb vs VSERVER_GSLB2 http
Done
>add gslb vs VSERVER_GSLB_BACKUP1 http
Done
>set gslb vs VSERVER_GSLB1 -backupvserver VSERVER_GSLB_BACKUP1
Done
>add gslb service SERVICE_GSLB1 1.1.1.1 HTTP 80 -sitename site1
Done
>add gslb service SERVICE_GSLB2 1.1.1.2 HTTP 80 -sitename site1
Done
>add gslb service SERVICE_GSLB3 1.1.1.3 HTTP 80 -sitename site2
Done
>add gslb service SERVICE_GSLB4 1.1.1.4 HTTP 80 -sitename site2
Done
>bind gslb vs VSERVER_GSLB1 -servicename SERVICE_GSLB1
Done
>bind gslb vs VSERVER_GSLB_BACKUP1 -servicename SERVICE_GSLB2
Done
>bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB3
Done
```

```
>bind gslb vs VSERVER_GSLB2 -servicename SERVICE_GSLB4
Done
>add cs action a1 -targetvserver VSERVER_GSLB1
Done
>add cs policy p1 -rule "CLIENT.IP.SRC.EQ(5.5.5.5)" -action a1
Done
>bind cs vs CSVSERVER_GSLB -domainName www.abc.com
Done
>bind cs vs CSVSERVER_GSLB -policyname p1 -priority 1
Done
>add cs action a2 -targetvserver VSERVER_GSLB2
Done
>add cs policy p2 -rule "CLIENT.IP.SRC.EQ(6.6.6.6)" -action a2
Done
>bind cs vs CSVSERVER_GSLB -policyname p2 -priority 2
Done
```

Configuring Global Server Load Balancing for DNS Queries with NAPTR records

September 21, 2020

In a typical Global Server Load Balancing (GSLB) deployment, the NetScaler appliance receives DNS queries for A/AAAA records, selects the most appropriate GSLB service according to the configured load balancing method, and returns the service's IP address as a reply to the DNS query. You can now configure the appliance to receive DNS queries for NAPTR records and respond with the list of services configured for a domain. The appliance also monitors the health of the services, and in the response it provides a list of only the services that are up.

Example:

In Telco deployments, you can configure a NetScaler appliance to receive DNS queries with NAPTR records from clients such as mobile management entities (MMEs), which play the role of a DNS resolver to discover all the services that are offered by the domain name. The appliance responds to the query with NAPTR records for all the services that are up. The MME can use this NAPTR response to run the S-NAPTR procedure to select the nodes on the basis of the service offered, colocation, topological closeness, and so on.

If multiple nodes qualify for selection, the MME can use the preference field in the NAPTR record from the NetScaler appliance to determine the node.

NAPTR Record Format

While responding to a DNS query with NAPTR record, a NetScaler appliance constructs a response NAPTR record for each GSLB service.

The following table lists the files in the NAPTR record:

Field	
Domain	The GSLB domain
TTL	The amount of time for which the NAPTR record can be cached.
Class	The class of the record. By default, this value is set to IN.
Type	The DNS record type.
Order	Specifies the order in which the NAPTR record MUST be processed. You can specify the order in the GSLB service. Otherwise, it is set to 1.
Preference	Specifies the order in which NAPTR records with equal "order" values SHOULD be processed, low numbers being processed before high numbers. If the order is not specified in the GSLB service, it is set to 1.
Flags	Controls the aspects of the rewriting and interpretation of the fields in the record. The NetScaler appliance sets this value to A.
Service	Specifies the available service(s).
Regular Expression	Regular expressions are not supported, so this value is set to NULL.
Replacement	The domain name of the node that hosts the services.

Configuration procedure

For detailed GSLB configuration instructions, see Configuring Global Server Load Balancing (GSLB). Make sure that you do the following:

- Set the following parameters while adding the GSLB virtual server:
  - serviceType: ANY

  - dnsRecordType: NAPTR
  - lbMethod: CUSTOMLOAD

Example:

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD
<!--NeedCopy-->
```

- While adding a GSLB site, set the *naptrReplacementSuffix* parameter to the domain name that you want to embed in the NAPTR records.

Example:

```
add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com
<!--NeedCopy-->
```

- Set the following parameters while adding the GSLB service:
  - naptrreplacement
  - naptrOrder
  - naptrServices
  - naptrDomainTTL
  - naptrPreference

Sample configuration

```
add gslb vserver gslb_vs ANY -dnsRecordType NAPTR -lbMethod CUSTOMLOAD

Done

add gslb site site1 10.102.218.200  -naptrReplacementSuffix example.com

Done

add gslb service sgw1 3.3.3.13 ANY *  -siteName site1 -naptrreplacement sgw1.site1. -naptrOrder 2 -naptrServices x-3gpp-sgw:x-s5-gtp -naptrDomainTTL 20 -naptrPreference   200

Done

add gslb service sgw2 3.3.3.11 ANY *  -siteName site1 -naptrreplacement sgw2.site1. -naptrOrder 5 -naptrServices x-3gpp-sgw:x-s5-gtp  -naptrDomainTTL 20  naptrPreference  100

Done
```

```

add gslb service sgw3 3.3.3.12 ANY *  -siteName site2 -naptrreplacement sgw3.site1. -naptrOrder 10 -naptrServices x-3gpp-sgw:x-s5-gtp  -naptrDomainTTL 20 naptrPreference   300

bind gslb vserver gslb_vs -serviceName sgw1

Done

bind gslb vserver gslb_vs -serviceName sgw2

Done

bind gslb vserver gslb_vs -serviceName sgw3

Done

bind gslb service sgw1 -monitorName ping

Done

bind gslb service sgw2 -monitorName ping

Done

bind gslb service sgw3 -monitorName ping

Done

bind gslb vserver gslb_vs -domainName gslb.com -TTL 5

Done
<!--NeedCopy-->
```

> **Note**
>
> DNS queries with NAPTR records are not supported in parent-child configuration.

Using the EDNS0 Client Subnet Option for Global Server Load Balancing

September 21, 2020

EDNS Client Subnet (ECS) is a DNS header extension that provides the client subnet details. You can use these details to improve the accuracy of NetScaler Global Server Load Balancing (GSLB) by using the client network location rather than the DNS resolver location to determine the topological closeness of the client.

> **Note**
>
> NetScaler supports only EDNS0.

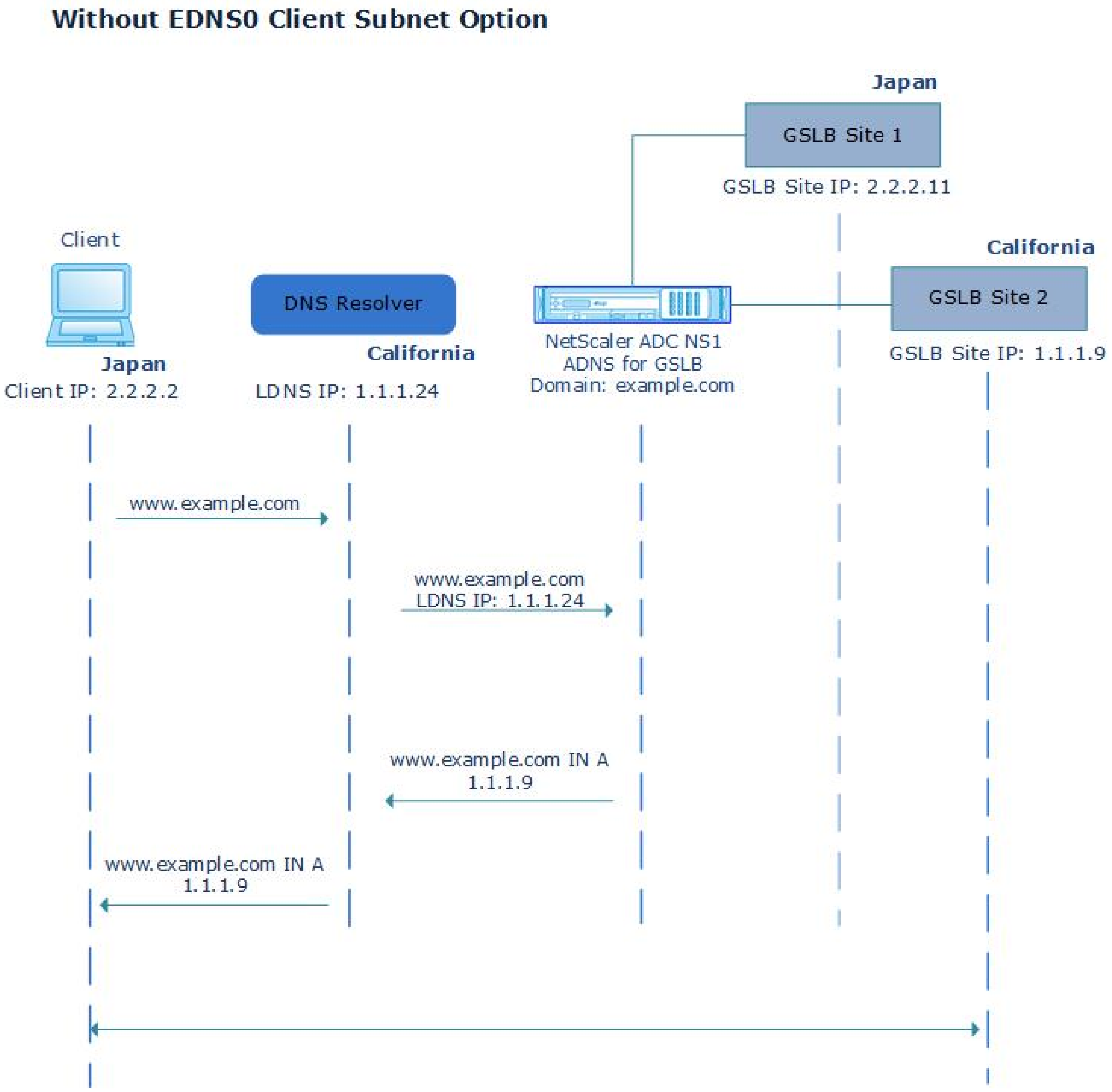
Important:

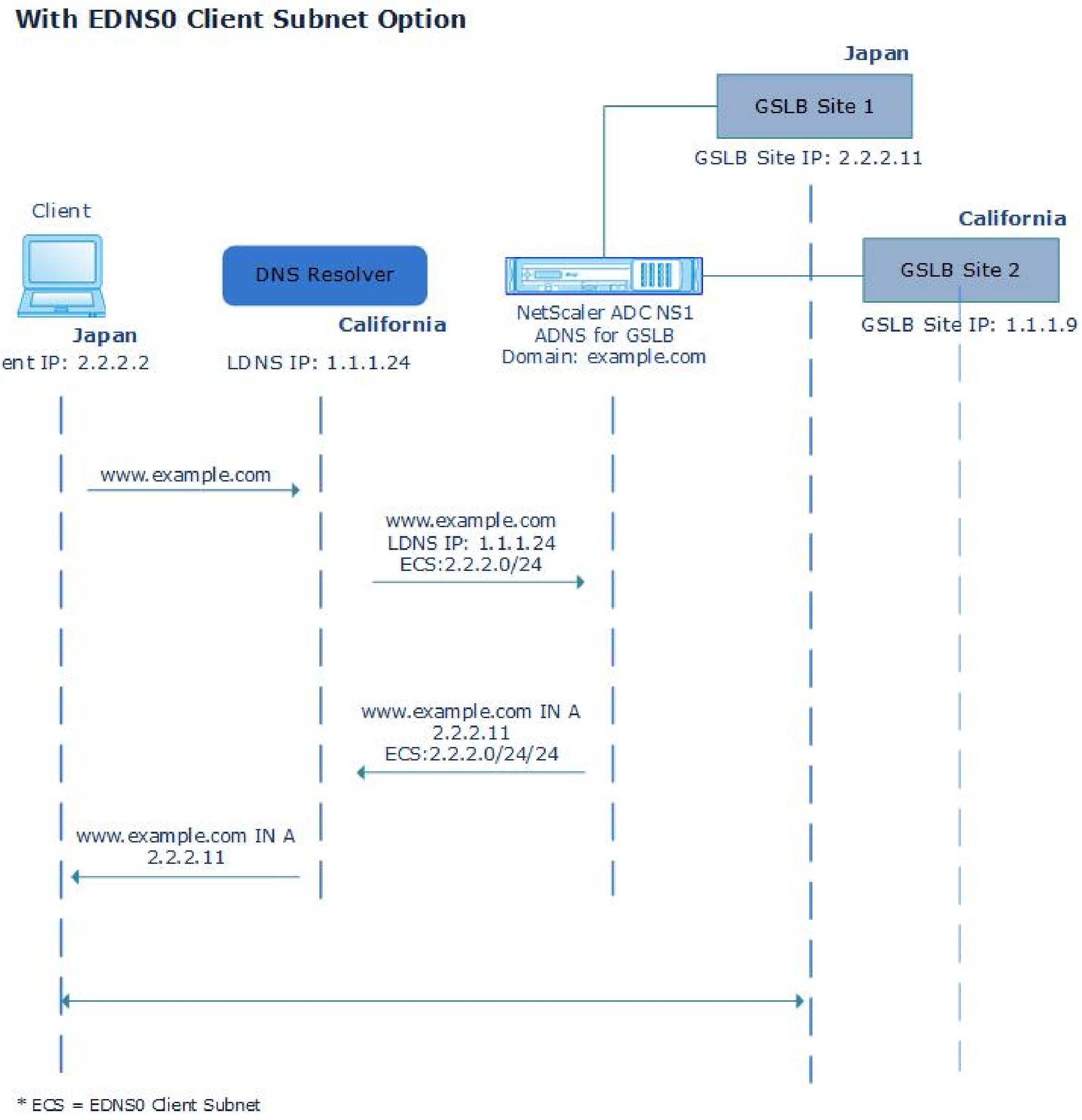
> Make sure that the LDNS in your deployment supports EDNS0 Client Subnet so that the incoming DNS queries contains the EDNS0 Client Subnet option and the NetScaler appliance uses the ECS address while processing the DNS query.

In a typical GSLB deployment, when you use proximity-based load balancing methods like static proximity or dynamic round-trip time (RTT), the NetScaler appliance uses the local DNS (LDNS) IP address for determining the topological closeness of the client and performs GSLB accordingly. But when a centralized DNS resolver, such as Google DNS or OpenDNS, is involved in the deployment, the NetScaler appliance sends the DNS request to a datacenter close to the centralized DNS resolver, which might not be close to the client. For example, in a typical NetScaler GSLB deployment using the static proximity load balancing method, an end-user request from Japan is sent to a datacenter in Japan and an end user request from California is sent to a datacenter in California. But if a centralized DNS resolver is involved, the NetScaler appliance might send a request from Japan to a datacenter in California.

You can use the ECS option in deployments that include the NetScaler appliance configured as Authoritative DNS (ADNS) server for a GSLB domain. If you use static proximity as the load balancing method, you can use the IP subnet in the EDNS header instead of the LDNS IP address to determine the geographical proximity of the client. In the case of proxy mode deployment, the NetScaler appliance forwards an ECS-enabled DNS query as-is to the back-end servers, and the appliance does not cache ECS-enabled DNS responses.

> **Note**
>
> The ECS option is not applicable for all other deployment modes, such as ADNS mode for non-GSLB domains, resolver mode, and forwarder mode. In all these modes, the ECS option is ignored by the NetScaler appliance. Also, by default, ECS is disabled for GSLB deployment.


Without EDNS0 Client Subnet Option
Japan
GSLB Site 1
GSLB Site IP: 2.2.2.11
Client
California
DNS Resolver
GSLB Site 2
NetScaler ADC NS1
ADNS for GSLB
Domain: example.com
GSLB Site IP: 1.1.1.9
Japan
Client IP: 2.2.2.2
California
LDNS IP: 1.1.1.24
www.example.com
www.example.com
LDNS IP: 1.1.1.24
www.example.com IN A
1.1.1.9
www.example.com IN A
1.1.1.9


With EDNS0 Client Subnet Option
Japan
GSLB Site 1
GSLB Site IP: 2.2.2.11
Client
California
GSLB Site 2
DNS Resolver
Japan
Client IP: 2.2.2.2
California
LDNS IP: 1.1.1.24
NetScaler ADC NS1
ADNS for GSLB
Domain: example.com
GSLB Site IP: 1.1.1.9
www.example.com
www.example.com
LDNS IP: 1.1.1.24
ECS:2.2.2.0/24
www.example.com IN A
2.2.2.11
ECS:2.2.2.0/24/24
www.example.com IN A
2.2.2.11
* ECS = EDNS0 Client Subnet

To enable EDNS0 Client Subnet option by using the command line interface:

At the command prompt, type:

```
set gslb vserver <vserver_name> **-ECS ENABLED

set gslb vserver vserver-GSLB-1 -ECS ENABLED
<!--NeedCopy-->
```

Address validation

You can configure a GSLB virtual server to verify that the address returned by the EDNS0 Client Subnet (ECS) option of the DNS query is not a private or an unroutable IP address. With address validation enabled, the NetScaler appliance ignores the ECS address in the DNS query if it is listed in the following table, and instead uses the LDNS IP address for global server load balancing.

> **Note**
>
> By default, address validation is disabled.

Address Type	Address	Description
IPV4	10.0.0.0/8	For private use
	172.16.0.0/12	For private use
	192.168.0.0/16	For private use
	0.0.0.0/8	Refers to the host on the network
	100.64.0.0/10	Shared address space
	127.0.0.0/8	Loopback address
	169.254.0.0/16	Link Local IPv4 address as defined in RFC 3927
	192.0.0.0/24	Used for IETF protocol assignments, includes the private space 192.168.0.0/16
	192.0.2.0/24	Used for documentation purposes
	192.88.99.0/24	Used for 6to4 Relay Anycast
	198.18.0.0/15	Used in Device benchmark testing
	198.51.100.0/24	Used for documentation purposes
	203.0.113.0/24	Used for documentation purposes
	240.0.0.0/4	Used as reserved
	255.255.255.255/32	Used for broadcast

IPv6	::1/128	loopback address
	::/128	unspecified address
	::ffff:0:0/96	IPv4-mapped address
	100::/64	discard-only address block
	2001::/23	Used for IETF protocol assignments
	2001::/32	TEREDO
	2001:2::/48	Used for benchmarking
	2001:db8::/32	Used for documentation purposes
	2001:10::/28	ORCHID
	2002::/16	Used for 6to4 Relay Anycast
	fc00::/7	Unique-local
	fe80::/10	Link-local Unicast addresses

To enable address validation by using the command line interface

At the command prompt, type:

```
set gslb vserver <vserver_name> -ecsAddressValidation ENABLED

set gslb vserver vserver-GSLB-1 -ecsAddressValidation ENABLED
<!--NeedCopy-->
```

Link load balancing

September 25, 2020

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the NetScaler appliance to monitor and control traffic so that packets are transmitted seamlessly over the best possible link. Unlike with server load balancing, where a service represents a server, with LLB, a service represents a router or the next hop. A link is a connection between the NetScaler appliance and the router.

To configure link load balancing, many users begin by configuring a basic setup with default settings. Configuring a basic setup involves configuring services, virtual servers, monitors, routes, an LLB method, and, optionally, configuring persistence. Once a basic setup is operational, you can customize it for your environment.

Load balancing methods that are applicable to LLB are round robin, destination IP hash, least bandwidth, and least packets. You can optionally configure persistence for connections to be sustained on a specific link. The available persistence types are source IP address-based, destination IP address-based, and source IP and destination IP address-based. PING is the default monitor but configuring a transparent monitor is recommended.

You can customize your setup by configuring reverse NAT (RNAT) and backup links.

Configuring a Basic LLB Setup

September 21, 2020

To configure LLB, you first create services representing each router to the Internet Service Providers (ISPs). A PING monitor is bound by default to each service. Binding a transparent monitor is optional but recommended. Then, you create a virtual server, bind the services to the virtual server, and configure a route for the virtual server. The route identifies the virtual server as the gateway to the physical routers represented by the services. The virtual server selects a router by using the load balancing method that you specify. Optionally, you can configure persistence to make sure that all traffic for a particular session is sent over a specific link.

To configure a basic LLB setup, do the following:

- Configure services
- Configure an LLB virtual server and binding a service
- Configure the LLB method and persistence
- Configure an LLB route
- Create and bind a transparent monitor

Configure services

A default monitor (PING) is automatically bound to a service type of ANY when the service is created, but you can replace the default monitor with a transparent monitor, as described in “
Creating and Binding a Transparent Monitor.”

To create a service by using the command line interface

At the command prompt, type:

```
add service <name> <IP> <serviceType> <port>

show service <name>
<!--NeedCopy-->
```

Example:

```
add service ISP1R_svc_any 10.10.10.254 any *
show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: DOWN
        Last state change was at Tue Aug 31 04:31:13 2010
        Time since last state change: 2 days, 05:34:18.600
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: ping
                State: UP       Weight: 1
                Probes: 244705  Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

To create services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Create Service dialog box, specify values for the following parameters:
    - Service Name*—name
    - Server—IP
    - Protocol*—serviceType (Select ANY from the drop-down list.)
    - Port*—port

A required parameter

1. Click Create.
2. Repeat Steps 2-4 to create another service.
3. Click Close.
4. In the Services pane, select the services that you just configured and verify that the settings displayed at the bottom of the screen are correct.

Configure an LLB virtual server and bind a service

After you create a service, create a virtual server and bind services to the virtual server. The default LB method of least connections is not supported in LLB. For information about changing the LB method, see “
Configuring the LLB Method and Persistence.”

To create a link load balancing virtual server and bind a service by using the command line interface

At the command prompt, type:

```
add lb vserver <name> <serviceType>

bind lb vserver < name> <serviceName>

show lb vserver < name>
<!--NeedCopy-->
```

Example:

```
add lb vserver LLB-vip any
bind lb vserver LLB-vip ISP1R_svc_any
```

```
sh lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Thu Sep  2 10:51:32 2010
        Time since last state change: 0 days, 17:51:46.770
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: NONE
        Connection Failover: DISABLED

1) ISP1R_svc_any (10.10.10.254: *) - ANY State: DOWN    Weight: 1
 Done
<!--NeedCopy-->
```

To create a link load balancing virtual server and bind a service by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
2. In the **IP Address Type** drop-down list, select the desired option. Select **Non Addressable** to create a virtual server that is not directly accessible.
3. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

Configure the LLB method and persistence

By default, the NetScaler appliance uses the least connections method to select the service for redirecting each client request, but you should set the LLB method to one of the supported methods. You can also configure persistence, so that different transmissions from the same client are directed to the same server.

To configure the LLB method and/or persistence by using the command line interface

At the command prompt, type the following command:

```
set lb vserver <name> -lbMethod <lbMethod> -persistenceType <
   persistenceType>
```

```

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver LLB-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver LLB-vip
        LLB-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: DOWN
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 00:52:21.200
        Effective State: DOWN
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: SOURCEIP
     Persistence Mask: 255.255.255.255       Persistence v6MaskLength: 128   Persistence Timeout: 2 min
        Connection Failover: DISABLED
<!--NeedCopy-->
```

To configure the link load balancing method and/or persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server for which you want to configure the load balancing method and/or persistence settings.
2. In the **Advanced Settings** section, select Method and configure the load balancing method.
3. In the **Advanced Settings** section, select **Persistence** and configure the persistence parameters.

Configure an LLB route

After configuring the
IPv4 or IPv6 services, virtual servers, LLB methods, and persistence, you configure an
IPv4 or IPv6 LLB route for the network specifying the LLB virtual server as the gateway. A route is a collection of links that are load balanced. Requests are sent to the LLB virtual server IP address that acts as the gateway for all outbound traffic and selects the router based on the LLB method configured.

To configure an IPv4 LLB route by using the command line interface

At the command prompt, type:

```
add lb route <network> <netmask> <gatewayName>

show lb route [<network> <netmask>]
<!--NeedCopy-->
```

Example:

```
add lb route 0.0.0.0 0.0.0.0 LLB-vip
show lb route 0.0.0.0 0.0.0.0
       Network          Netmask        Gateway/VIP             Flags
     -----------      -------------   --------------          --------
1)     0.0.0.0          0.0.0.0        LLB-vip               UP
<!--NeedCopy-->
```

To configure an IPv6 LLB route by using the command line interface

At the command prompt, type:

```
add lb route6 <network> <gatewayName>

show lb route6
<!--NeedCopy-->
```

Example:

```
add lb route6 ::/0 llb6_vs show lb route6 Network VIP Flags ----------- --------- -------- 1) ::/0 llb6_vs UP
<!--NeedCopy-->
```

To configure an LLB route by using the configuration utility

Navigate to System > Network > Routes, and select **LLB**, and configure the LLB route.

Note: Select LLBV6 to configure an IPV6 route.

To configure an LLB route by using the configuration utility

1. Navigate to System > Network > Routes.
2. In the details pane, select one of the following:

  - Click LLB to configure an IPv4 route.
  - Click LLBV6 to configure an IPv4 route.

3. In the Create LB Route or Create LB IPV6 Routedialog box, set the following parameters:

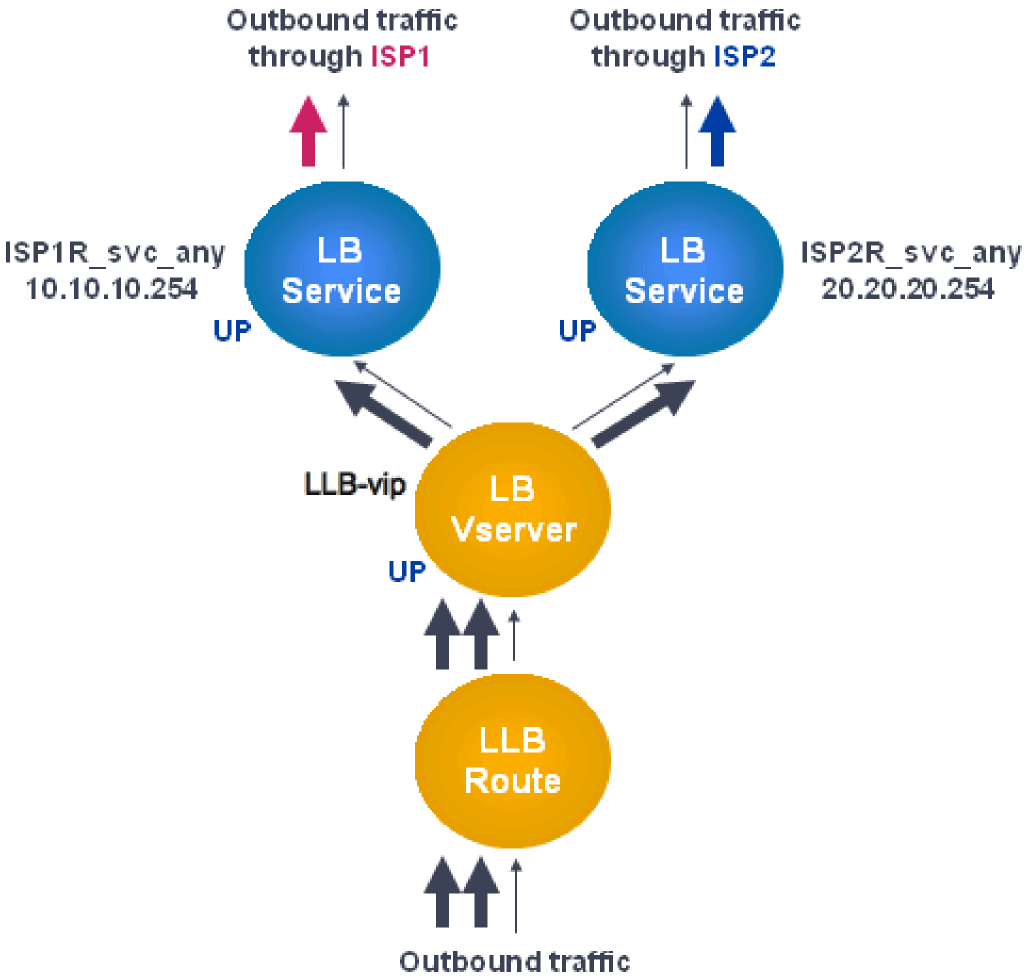
   - Network*
   - Netmask*—Required for IPV4 routes.
   - Gateway Name*—gatewayName

A required parameter

1. Click Create, and then click Close. The route that you just created appears on the LLB or the LLB6 tab in the Routes pane.

The following diagram shows a basic LLB setup. A service is configured for each of the two links (ISPs) and PING monitors are bound by default to these services. A link is selected based on the LLB method configured.

Figure 1. Basic LLB Setup


Outbound traffic through ISP1
Outbound traffic through ISP2
ISP1R_svc_any
10.10.10.254
LB Service
UP
LB Service
UP
ISP2R_svc_any
20.20.20.254
LLB-vip
LB Vserver
UP
LLB Route
Outbound traffic

> **Note**
>
> If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

Create and bind a transparent monitor

You create a transparent monitor to monitor the health of upstream devices, such as routers. You can then bind the transparent monitor to services. The default PING monitor monitors the connectivity only between the NetScaler appliance and the upstream device. The transparent monitor monitors all the devices existing in the path from the appliance to the device that owns the destination IP address specified in the monitor. If a transparent monitor is not configured and the status of the router is UP but one of the next hop devices from that router is down, the appliance includes the router while performing load balancing and forwards the packet to the router. However, the packet is not delivered to the final destination because one of the next hop devices is down. By binding a transparent monitor, if any of the devices (including the router) are down, the service is marked as DOWN and the router is not included when the appliance performs link load balancing.

To create a transparent monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent YES

show lb monitor [<monitorName>]
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-1 PING -destIP 10.10.10.11 -transparent YES
> show lb monitor monitor-1
1)    Name.......: monitor-1  Type......:      PING   State....: ENABLED
Standard parameters:
  Interval.........:            5 sec   Retries...........:                 3
  Response timeout.:            2 sec   Down time.........:            30 sec
  Reverse..........:               NO   Transparent.......:               YES
  Secure...........:               NO   LRTM..............:           ENABLED
```

```
 Action...........:   Not applicable   Deviation.........:
    0 sec
 Destination IP...:      10.10.10.11
 Destination port.:    Bound service
 Iptunnel.........:               NO
 TOS..............:               NO   TOS ID............:
                  0
 SNMP Alert Retries:                0     Success Retries..:
                  1
 Failure Retries..:                 0
<!--NeedCopy-->
```

To create a transparent monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors and configure a transparent monitor.

To create a transparent monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. In the Monitors pane, click Add.
3. In the Create Monitor dialog box, set the following parameters:
    - Name*
    - Type*
    - Destination IP
    - Transparent

A required parameter

1. Click Create, and then click Close.
2. In the Monitors pane, select the monitor that you just configured and verify that the settings displayed in the Details pane are correct.

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. On the **Monitors** tab, under **Available**, select the monitor that you want to bind to the service, and then click **Add**.

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind lb monitor <monitorName> <serviceName>

show service <name>
<!--NeedCopy-->
```

Example:

```
bind lb monitor monitor-HTTP-1 ISP1R_svc_any
 Done
> show service ISP1R_svc_any
        ISP1R_svc_any (10.10.10.254:*) - ANY
        State: UP
        Last state change was at Thu Sep  2 10:51:07 2010
        Time since last state change: 0 days, 18:41:55.130
        Server Name: 10.10.10.254
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): YES
        HTTP Compression(CMP): NO
        Idle timeout: Client: 120 sec   Server: 120 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED

1)      Monitor Name: monitor-HTTP-1
                State: UP  Weight: 1
                Probes: 1256       Failed [Total: 0 Current: 0]
                Last response: Success - ICMP echo reply received.
                Response Time: 1.322 millisec
 Done
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select a service to which you want to bind a monitor, and then click Open.
3. In the Configure Service dialog box, on the Monitors tab, under Available, select the monitor that you want to bind to the service, and then click Add.

4. Click OK.
5. In the Services pane, select the service that you just configured and verify that the settings displayed in the Details pane are correct.

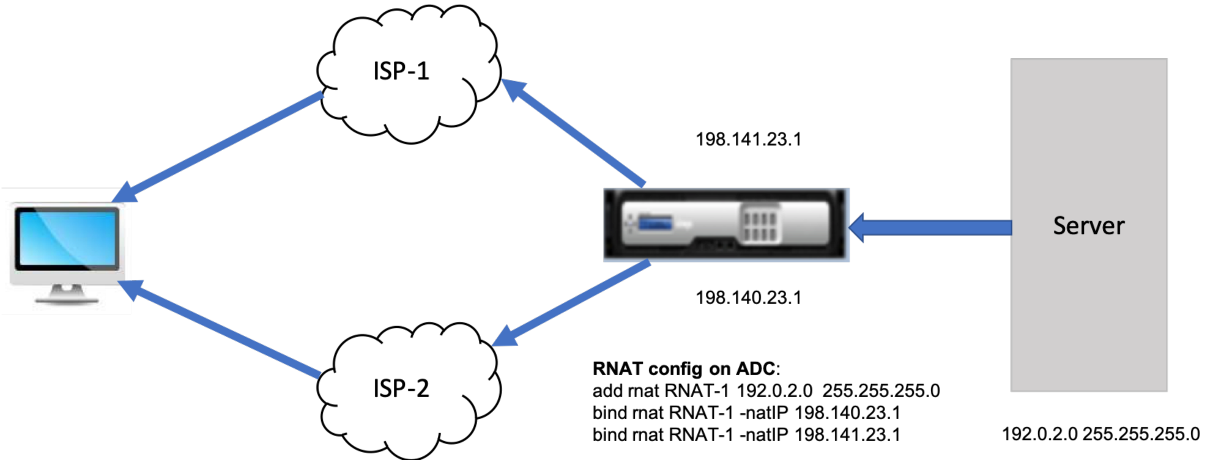
Configuring RNAT with LLB

September 21, 2020

To configure RNAT by using the command line interface

You can configure an LLB setup for reverse network address translation (RNAT) for outbound traffic. This ensures that the return network traffic for a specific flow is routed through the same path. First configure basic LLB, as described in
"Configuring a Basic LLB Setup", and then configure RNAT. You must then enable use subnet IP (USNIP) mode.

In the following diagram, the NetScaler appliance uses LLB to route outbound traffic to different links. During the RNAT operation, the NetScaler appliance replaces the source IP addresses of the outbound traffic with the public NAT IP address (198.141.23.1) to route the traffic through ISP-1. Similarly, the NetScaler appliance replaces the source IP addresses with 198.140.23.1 to route the traffic through ISP-2.



RNAT config on ADC:

```
add rnat RNAT-1 192.0.2.0  255.255.255.0
bind rnat RNAT-1 -natIP 198.140.23.1
bind rnat RNAT-1 -natIP 198.141.23.1
```

To add SNIPs for ISP routers by using the command line interface

At the command prompt, type:

```
add NS IP <subnet of first ISP in the IP router> <subnet mask> -type
    SNIP

```

```
add NS IP <subnet of second ISP in the IP router> <subnet mask> -type SNIP
<!--NeedCopy-->
```

Example:

```
add ns ip 10.140.23.1 255.255.255.0 -type snip

add ns ip 10.141.23.1 255.255.255.0 -type snip
<!--NeedCopy-->
```

To configure RNAT by using the command line interface

At the command prompt, type:

```
set rnat <network> <netmask>

show rnat
<!--NeedCopy-->
```

Example:

```
set rnat 10.102.29.0 255.255.255.0 -natIP 10.140.23.1
set rnat 10.102.29.0 255.255.255.0 -natIP 10.141.23.1
show rnat
    1) Network: 10.102.29.0      Netmask: 255.255.255.0
     NatIP: 10.140.23.1
     2) Network: 10.102.29.0      Netmask: 255.255.255.0
     NatIP: 10.141.23.1
<!--NeedCopy-->
```

To configure RNAT by using the configuration utility

1. Navigate to **System** > **Network** > **NATs**.
2. On the **RNAT** tab, click **Configure RNAT**.
3. Specify the network on which to perform RNAT.

> **Note**
>
> You can also configure RNAT by using Access Control Lists (ACLs). Refer Configuring RNAT for details.

To enable Use Subnet IP mode by using the command line interface

At the command prompt, type:

```
enable ns mode USNIP

show ns mode
<!--NeedCopy-->
```

Example:

```
enable ns mode USNIP

show ns mode
        Mode                           Acronym              Status
        -------                        -------              ------
 1)     Fast Ramp                      FR                   ON
 2)     … .
 8)     Use Subnet IP                  USNIP                ON
 9)     …
<!--NeedCopy-->
```

To enable Use Subnet IP mode by using the configuration utility

1. Navigate to **System** > **Settings** and, under **Modes and Features**, click **Configure Modes**.
2. In the **Configure Modes** dialog box, select **Use Subnet IP**, and then click **OK**.

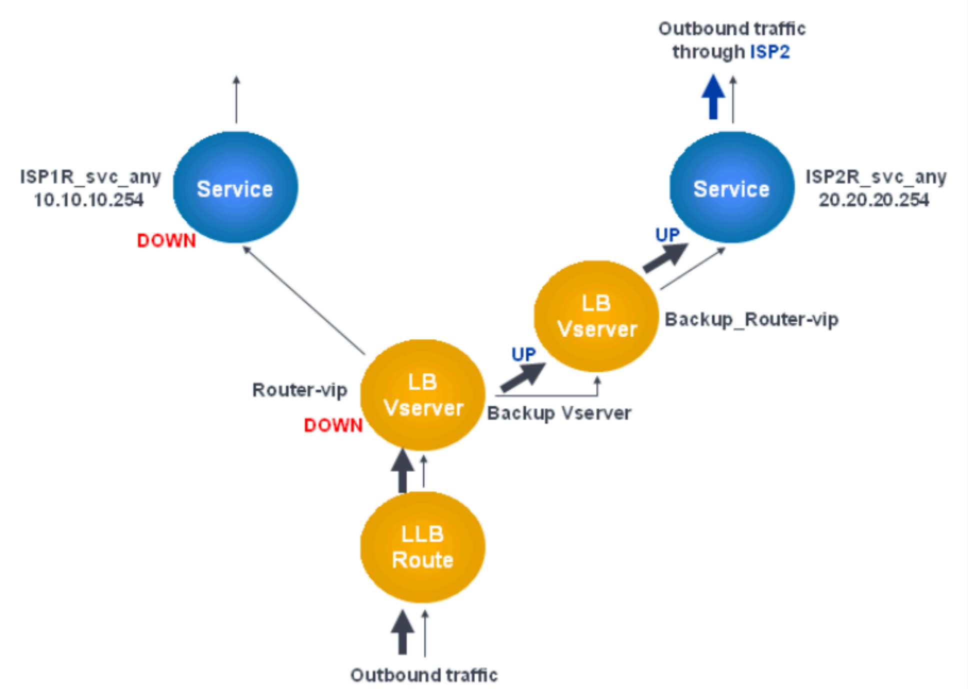
Configuring a Backup Route

September 21, 2020

To prevent disruption in services when the primary route is down, you can configure a backup route. Once the backup route is configured, the NetScaler appliance automatically uses it when the primary route fails. First create a primary virtual server as described in "Configuring an LLB Virtual Server and Binding a Service." To configure a backup route, create a secondary virtual server similar to a primary virtual server and then designate this virtual server as a backup virtual server (route).

In the following diagram, **Router-vip** is the primary virtual server, and **Backup_Router-vip** is the secondary virtual server designated as the backup virtual server.

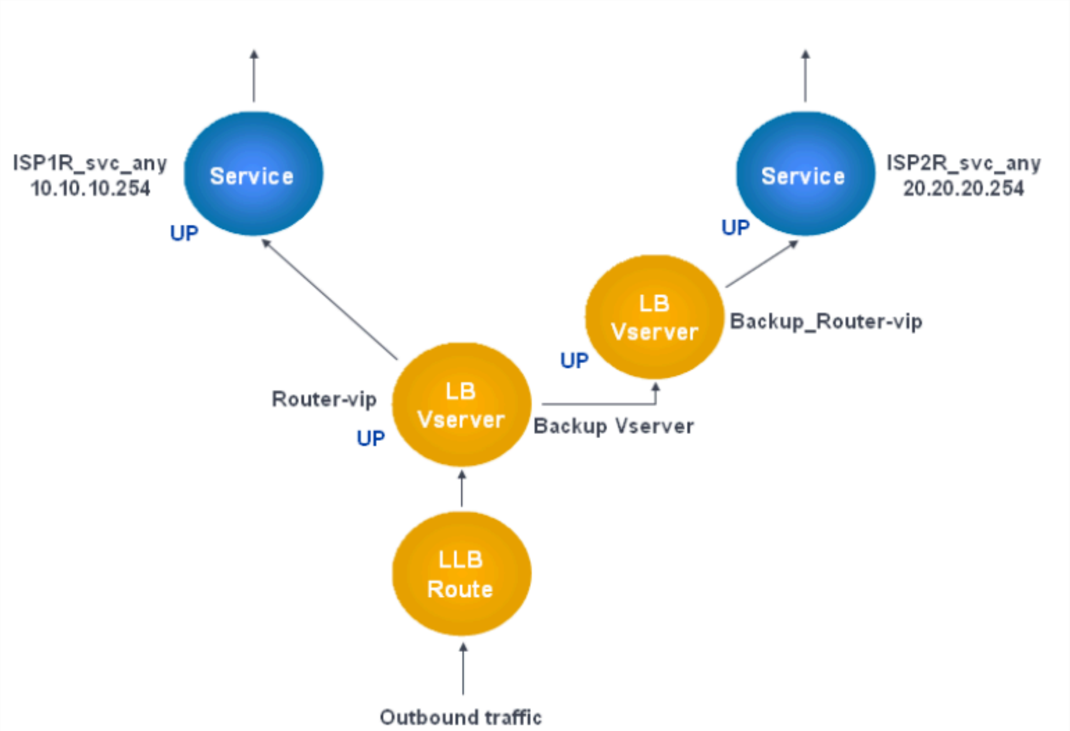
Figure 1. Backup Route Setup


Outbound traffic
through ISP2
ISP1R_svc_any
10.10.10.254
Service
DOWN
Service
ISP2R_svc_any
20.20.20.254
UP
LB
Vserver
Backup_Router-vip
UP
Router-vip
LB
Vserver
DOWN
Backup Vserver
LLB
Route
Outbound traffic

Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

By default, all traffic is sent through the primary route. However, when the primary route fails, all traffic is diverted to the backup route as shown in the following diagram.

Figure 2. Backup Routing in Operation


ISP1R_svc_any
10.10.10.254
Service
UP
Service
ISP2R_svc_any
20.20.20.254
UP
LB
Vserver
Backup_Router-vip
UP
Router-vip
LB
Vserver
UP
Backup Vserver
LLB
Route
Outbound traffic

Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

To set the secondary virtual server as the backup virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -backupVserver <string>
<!--NeedCopy-->
```

Example:

```
set lb vserver Router-vip -backupVServer Backup_Router-vip
> show lb vserver Router-vip
        Router-vip (0.0.0.0:0) - ANY    Type: ADDRESS
        State: UP
        Last state change was at Fri Sep  3 04:46:48 2010
        Time since last state change: 0 days, 03:09:45.600
        Effective State: UP
        Client Idle Timeout: 120 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: ROUNDROBIN
        Mode: IP
        Persistence: DESTIP     Persistence Mask: 255.255.255.255          Persistence v6MaskLength: 128   Persistence Timeout: 2 min
        Backup: Router2-vip
        Connection Failover: DISABLED
 Done
<!--NeedCopy-->
```

To set the secondary virtual server as the backup virtual server by using the configuration utility

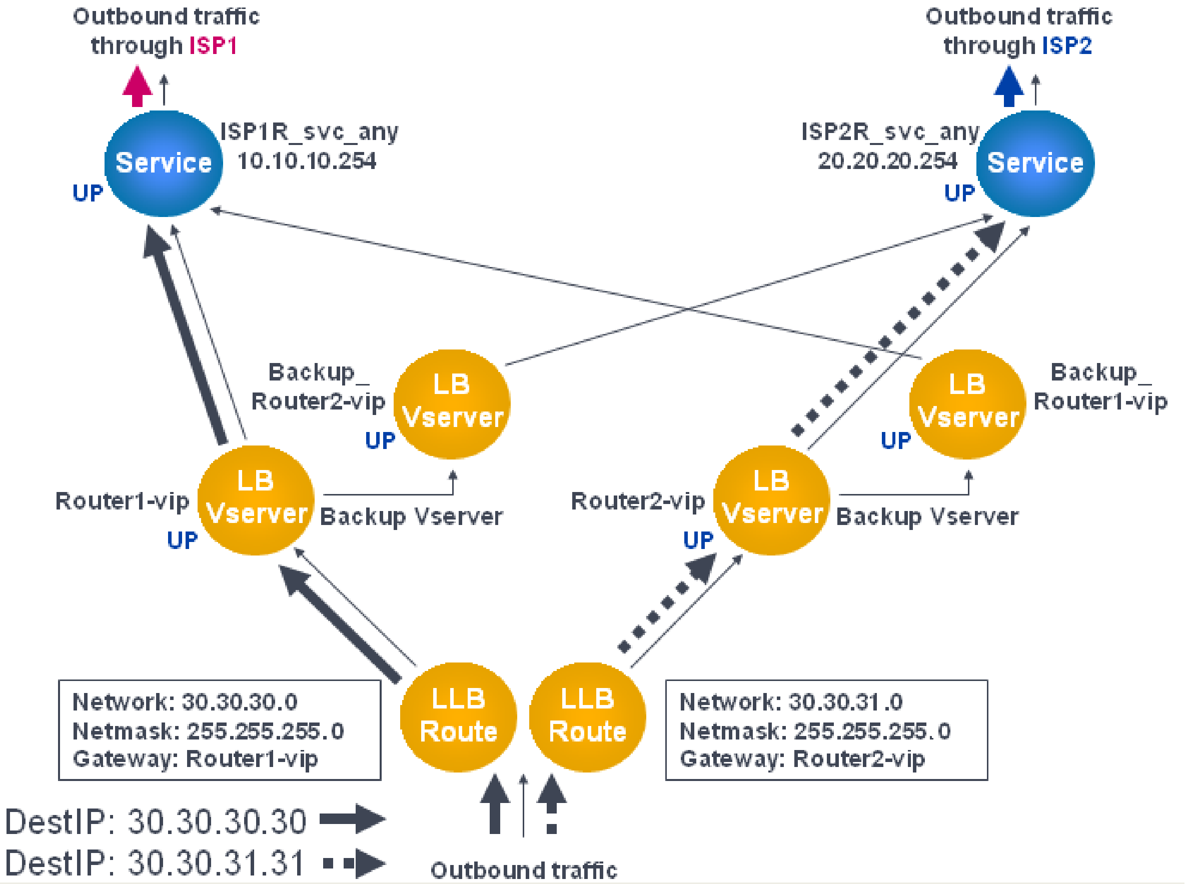
1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the secondary virtual server for which you want to configure the backup virtual server.
2. In the **Load Balancing Virtual Server** dialog box, under **Advanced**, select **Protection**.
3. In the **Backup Virtual Server** drop-down list, select the secondary backup virtual server, and then click **OK**.

Resilient LLB Deployment Scenario

September 21, 2020

In the following diagram, there are two networks: 30.30.30.0 and 30.30.31.0. Link load balancing is configured based on the destination IP address. Two routes are configured with gateways **Router1-vip** and **Router2-vip**, respectively. **Router1-vip** is configured as a backup to **Router2-vip** and vice versa. All traffic with the destination IP specified as 30.30.30.30 is sent through **Router1-vip** and traffic with the destination IP specified as 30.30.31.31 is sent through **Router2-vip**.

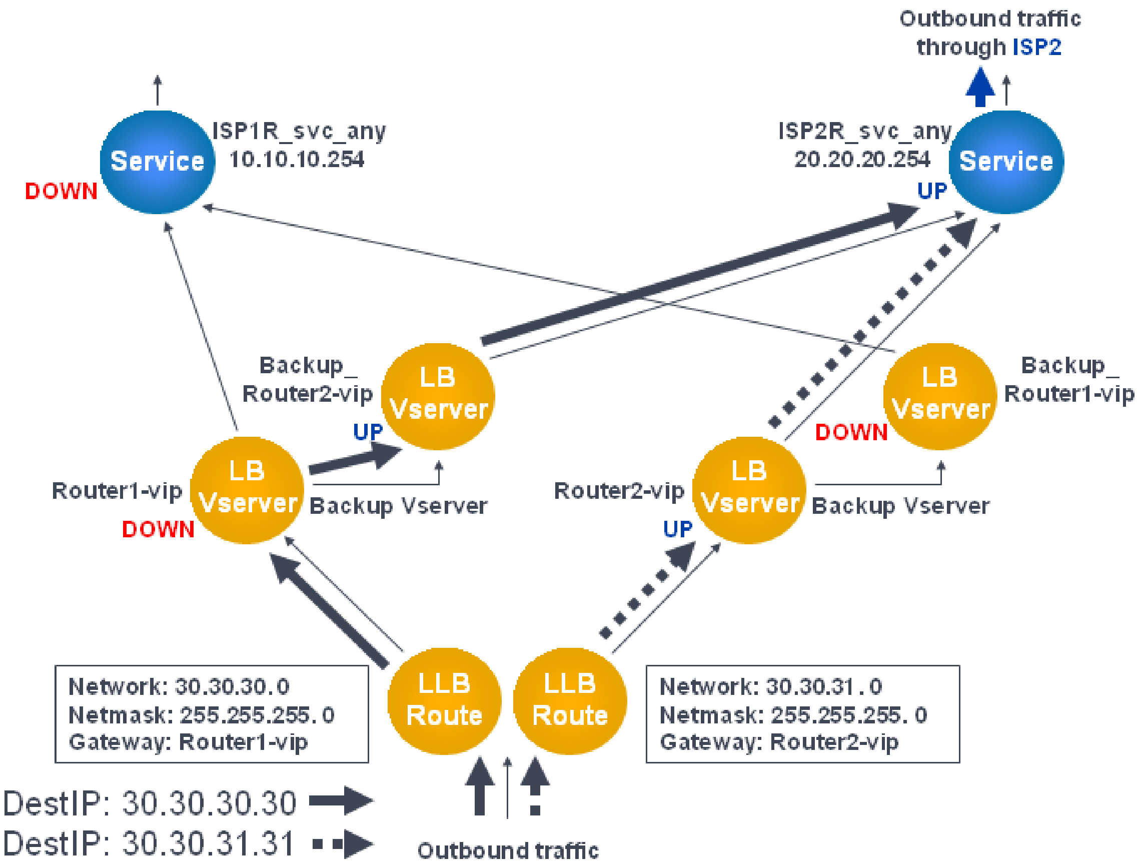
Figure 1. Resilient LLB Deployment Setup


Outbound traffic
through ISP1
Outbound traffic
through ISP2
ISP1R_svc_any
10.10.10.254
Service
UP
ISP2R_svc_any
20.20.20.254
Service
UP
Backup_
Router2-vip
LB
Vserver
UP
LB
Vserver
UP
Backup_
Router1-vip
Router1-vip
LB
Vserver
UP
Backup Vserver
Router2-vip
LB
Vserver
UP
Backup Vserver
Network: 30.30.30.0
Netmask: 255.255.255.0
Gateway: Router1-vip
LLB
Route
LLB
Route
Network: 30.30.31.0
Netmask: 255.255.255.0
Gateway: Router2-vip
DestIP: 30.30.30.30
DestIP: 30.30.31.31
Outbound traffic

Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

However, if any one of the gateways (**Router1-vip** or **Router2-vip**) is DOWN, traffic is routed through the backup router. In the following diagram, **Router1-vip** for ISP1 is DOWN, so all traffic with the destination IP specified as 30.30.30.30 is also sent through ISP2.

Figure 2. Resilient LLB Deployment Scenario


Outbound traffic
through ISP2
ISP1R_svc_any
10.10.10.254
Service
DOWN
ISP2R_svc_any
20.20.20.254
Service
UP
Backup_
Router2-vip
LB
Vserver
UP
Backup_
Router1-vip
LB
Vserver
DOWN
Router1-vip
LB
Vserver
DOWN
Backup Vserver
Router2-vip
LB
Vserver
UP
Backup Vserver
Network: 30.30.30.0
Netmask: 255.255.255.0
Gateway: Router1-vip
LLB
Route
LLB
Route
Network: 30.30.31.0
Netmask: 255.255.255.0
Gateway: Router2-vip
DestIP: 30.30.30.30
DestIP: 30.30.31.31
Outbound traffic

Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

Monitoring an LLB Setup

September 21, 2020

After the configuration is up and running, you should view the statistics for each service and virtual server to check for possible problems.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name

- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the command line interface

To display a summary of the statistics for all the virtual servers currently configured on the NetScaler, or for a single virtual server, at the command prompt, type:

```
stat lb vserver -detail] [<name>]
<!--NeedCopy-->
```

Example:

```
stat lb vserver -detail
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
                           Hits/s
One                        *    80         HTTP           UP      5/s
        0/s
Two                        *     0          TCP         DOWN      0/s
        0/s
Three                      *  2598          TCP         DOWN      0/s
        0/s
dnsVirtualNS    10.102.29.90    53          DNS         DOWN      0/s
        0/s
BRVSERV            10.10.1.1    80         HTTP         DOWN      0/s
        0/s
LBVIP           10.102.29.66    80         HTTP           UP      0/s
        0/s
 Done
<!--NeedCopy-->
```

To display virtual server statistics by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers > Statistics.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server, and click Statistics.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the command line interface

At the command prompt, type:

```
stat service <name>
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

Load Balancing

September 21, 2020

The load balancing feature distributes user requests for web pages and other protected applications across multiple servers that all host (or mirror) the same content. You use load balancing primarily to manage user requests to heavily used applications, preventing poor performance and outages and ensuring that users can access your protected applications. Load balancing also provides fault tolerance; when one server that hosts a protected application becomes unavailable, the feature distributes user requests to the other servers that host the same application.

You can configure the load balancing feature to:

- Distribute all requests for a specific protected website, application, or resource between two or more identically configured servers.
- Use any of several different algorithms to determine which server should receive each incoming user request, basing the decision on different factors, such as which server has the fewest current user connections or which server has the lightest load.

The load balancing feature is a core feature of the NetScaler appliance. Most users first set up a working basic configuration and then customize various settings, including persistence for connections. In addition, you can configure features for protecting the configuration against failure, managing client traffic, managing and monitoring servers, and managing a large scale deployment.

How load balancing works

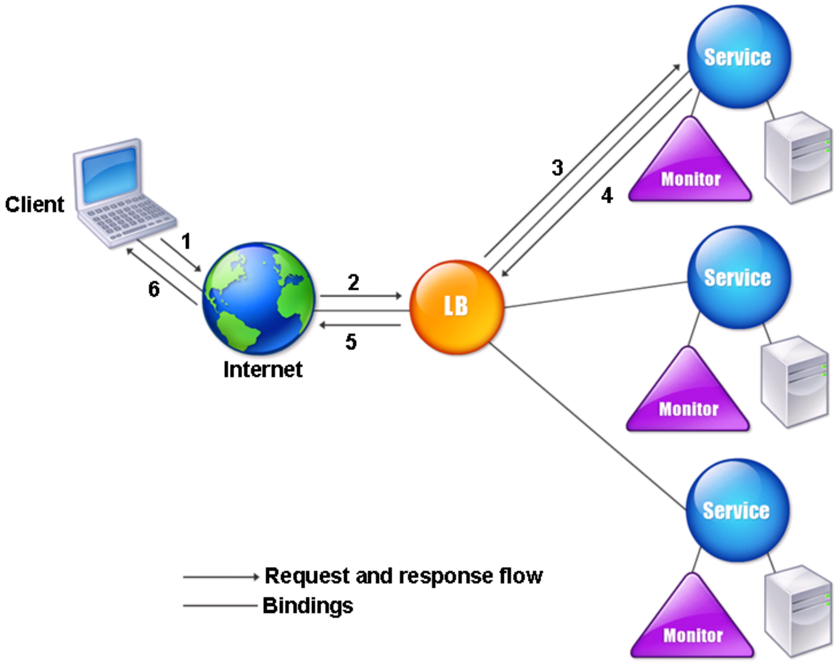
September 21, 2020

In a basic load balancing setup, clients send their requests to the IP address of a virtual server configured on the NetScaler appliance. The virtual server distributes them to the load-balanced application servers according to a preset pattern, called the load balancing algorithm. In some cases, you might want to assign the load balancing virtual server a wildcard address instead of a specific IP address. For instructions about specifying a global HTTP port on the appliance, see **Global HTTP Ports**.

Load balancing basics

A load balancing setup includes a load-balancing virtual server and multiple load-balanced application servers. The virtual server receives incoming client requests, uses the load balancing algorithm to select an application server, and forwards the requests to the selected application server. The following conceptual drawing illustrates a typical load balancing deployment. Another variation involves assigning a global HTTP port.

Figure 1. Load Balancing Architecture



The load balancing virtual server can use any of a number of algorithms (or methods) to determine how to distribute load among the load-balanced servers that it manages. The default load balancing method is the least connection method, in which the NetScaler appliance forwards each incoming client connection to whichever load-balanced application server currently has the fewest active user connections.

The entities that you configure in a typical NetScaler load balancing setup are:

- **Load balancing virtual server.** The IP address, port, and protocol combination to which a client sends connection requests for a particular load-balanced website or application. If the application is accessible from the Internet, the virtual server IP (VIP) address is a public IP address. If the application is accessible only from the local area network (LAN) or wide area network (WAN), the VIP is usually a private (ICANN non-routable) IP address.
- **Service.** The IP address, port, and protocol combination used to route requests to a specific load-balanced application server. A service can be a logical representation of the application server itself, or of an application running on a server that hosts multiple applications. After creating a service, you bind it to a load balancing virtual server.
- **Server object.** A virtual entity that enables you to assign a name to a physical server instead of identifying the server by its IP address. If you create a server object, you can specify its name instead of the server's IP address when you create a service. Otherwise, you must specify the

server's IP address when you create a service, and the IP address becomes the name of the server.
- **Monitor.** An entity on the NetScaler appliance that tracks a service and ensures that it is operating correctly. The monitor periodically probes (or performs a health check on) each service to which you assign it. If the service does not respond within the time specified by the time-out, and a specified number of health checks fail, that service is marked DOWN. The NetScaler appliance then skips that service when performing load balancing, until the issues that caused the service to quit responding are fixed.

The virtual server, services, and load balanced application servers in a load balancing setup can use either Internet Protocol version 4 (IPv4) or Internet Protocol version 6 (IPv6) IP addresses. You can mix IPv4 and IPv6 addresses in a single load balancing setup.

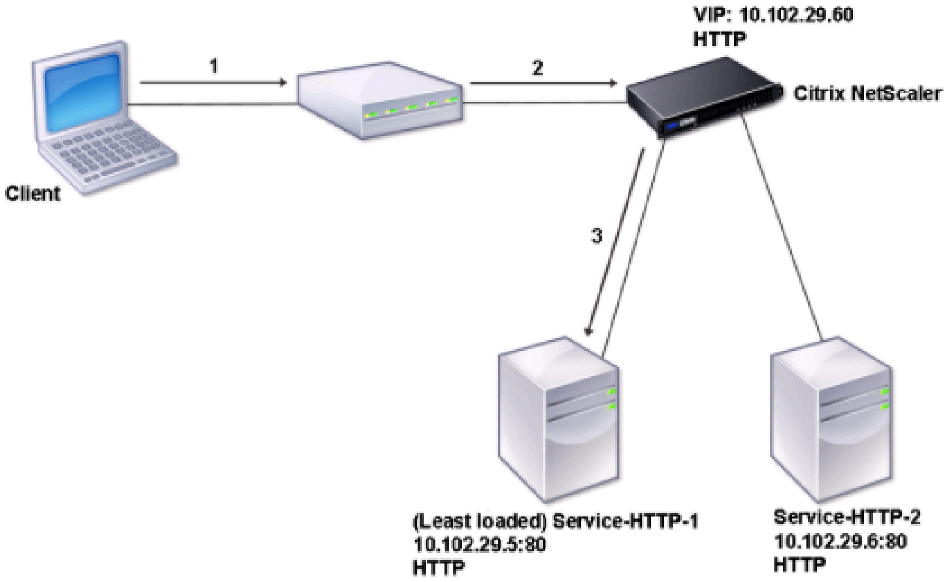
For variations in the load balancing setup, see the following use cases:

- Configuring Load Balancing in Direct Server Return Mode
- Configuring LINUX Servers in DSR Mode
- Configuring DSR Mode When Using TOS
- Configuring Load Balancing in DSR Mode by Using IP Over IP
- Configuring Load Balancing in One-arm Mode
- Configuring Load Balancing in the Inline Mode
- Load Balancing of Intrusion Detection System Servers
- Load balance remote desktop protocol servers

Understanding the topology

In a load balancing setup, the load balancing server is logically located between the client and the server farm, and manages traffic flow to the servers in the server farm. On the NetScaler appliance, the application servers are represented by virtual entities called services. The following diagram shows the topology of a basic load balancing configuration.

Figure 2. Basic Load Balancing Topology

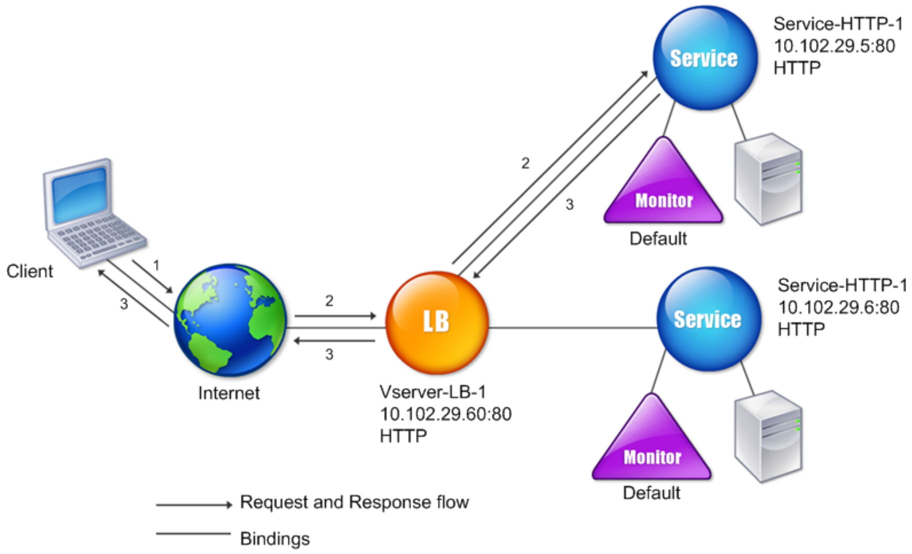

VIP: 10.102.29.60
HTTP
1
2
Citrix NetScaler
Client
3
(Least loaded) Service-HTTP-1
10.102.29.5:80
HTTP
Service-HTTP-2
10.102.29.6:80
HTTP

In the diagram, load balancing is used to manage traffic flow to the servers. The virtual server selects the service and assigns it to serve client requests. Consider a scenario where the services Service-HTTP-1 and Service-HTTP-2 are created and bound to the virtual server named Vserver-LB-1. Vserver-LB-1 forwards the client request to either Service-HTTP-1 or Service-HTTP-2. The NetScaler appliance uses the least connection load balancing method to select the service for each request. The following table lists the names and values of the basic entities that must be configured on the appliance.

Entity	Name	IPAddress	Port	Protocol
Virtual server	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	80	HTTP
	Service-HTTP-2	10.102.29.6	80	HTTP
Monitors	Default	None	None	None

The following diagram shows the load balancing sample values and mandatory parameters that are described in the preceding table.

Figure 3. Load Balancing Entity Model


Service
Service-HTTP-1
10.102.29.5:80
HTTP
Monitor
Default
2
3
Client
1
3
Internet
2
3
LB
Vserver-LB-1
10.102.29.60:80
HTTP
Service
Service-HTTP-1
10.102.29.6:80
HTTP
Monitor
Default
Request and Response flow
Bindings

Use of wildcards instead of IP addresses and ports

In some cases you might need to use a wildcard for the IP address or the port of a virtual server or for the port of a service. The following cases may require using a wildcard:

- If the NetScaler appliance is configured as a transparent pass through, which must accept all traffic that is sent to it regardless of the IP or port to which it is sent.
- If one or more services listen on ports that are not well known.
- If one or more services, over time, change the ports that they listen on.
- If you reach the limit for the number of IP addresses and ports that you can configure on a single NetScaler appliance.
- If you want to create virtual servers that listen for all traffic on a specific virtual LAN.

When a wildcard-configured virtual server or service receives traffic, the NetScaler appliance determines the actual IP address or port and creates new records for the service and associated load balanced application server. These dynamically created records are called dynamically learned server and service records.

For example, a firewall load balancing configuration can use wildcards for both the IP address and port. If you bind a wildcard TCP service to this type of load balancing virtual server, the virtual server receives and processes all TCP traffic that does not match any other service or virtual server.

The following table describes some of the different types of wildcard configurations and when each should be used.

IP	Port	Protocol	Description
*	*	TCP	A general wildcard virtual server that accepts traffic sent to any IP address and port on the NetScaler appliance. When using a wildcarded virtual server, the appliance dynamically learns the IP and port of each service and creates the necessary records as it processes traffic.
*	*	TCP	A firewall load balancing virtual server. You can bind firewall services to this virtual server, and the NetScaler appliance passes traffic through the firewall to the destination.

IP	Port	Protocol	Description
IP Address	*	TCP,UDP, and ANY	A virtual server that accepts all traffic that is sent to the specified IP address, regardless of the port. You must explicitly bind to this type of virtual server the services to which it will redirect traffic. It will not dynamically learn them. **Note**: You do not configure services or virtual servers for a global HTTP port. In this case, you configure a specific port as a global HTTP port (for example, set ns param -httpPort 80). The appliance then accepts all traffic that matches the port number, and processes it as HTTP traffic. The appliance dynamically learns and creates services for this traffic.

IP	Port	Protocol	Description
*	port	SSL, SSL_TCP	A virtual server that accepts all traffic sent to any IP address on a specific port. Used for global transparent SSL offloading. All SSL, HTTP, and TCP processing that usually is performed for a service of the same protocol type is applied to traffic that is directed to this specific port. The appliance uses the port to dynamically learn the IP of the service it should use. If –cleartext is not specified, the NetScaler appliance uses end-to-end SSL.
*	port	Not applicable	All other virtual servers that can accept traffic to the port. You do not bind services to these virtual servers; the NetScaler appliance learns them dynamically.

Note: If you have configured your NetScaler appliance as a transparent pass through that makes use of global (wildcard) ports, you may want to turn on Edge mode.
For more information, see “Configuring Edge Mode.”

The NetScaler appliance attempts to locate virtual servers and services by first attempting an exact match. If none is found, it continues to search for a match based on wildcards, in the following order:

1. Specific IP address and specific port number
2. Specific IP address and a * (wildcard) port
3. • (wildcard) IP address and a specific port
4. • (wildcard) IP address and a * (wildcard) port

If the appliance is unable to select a virtual server by IP address or port number, it searches for a virtual server on the basis of the protocol used in the request, in the following order:

1. HTTP
2. TCP
3. ANY

Configuring global HTTP ports

You do not configure services or virtual servers for a global HTTP port. Instead, you configure a specific port by using the set ns param command. After configuring this port, the NetScaler appliance accepts all traffic that matches the port number, and processes it as HTTP traffic, dynamically learning and creating services for that traffic.

You can configure more than one port number as a global HTTP port. If you are specifying more than one port number in a single set ns param command, separate the port numbers by a single white space. If one or more ports have already been specified as global HTTP ports, and you want to add one or more ports without removing the ports that are currently configured, you must specify all the port numbers, current and new, in the command. Before you add port numbers, use the show ns param command to view the ports that are currently configured.

To configure a global HTTP port by using the command line interface

At the command prompt, type the following commands to configure a global HTTP port and verify the configuration:

```
set ns param -httpPort <port>

show ns param
<!--NeedCopy-->
```

Example 1: Configuring a port as a global HTTP port

In this example, port 80 is configured as a global HTTP port.

```
set ns param -httpPort 80
Done
show ns param
         Global configuration settings:
                          HTTP port(s): 80
                       Max connections: 0
           Max requests per connection: 0
                   Client IP insertion: DISABLED
                        Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
        ...
        ...
<!--NeedCopy-->
```

Example 2: Adding ports when one or more global HTTP ports are already configured**

In this example, port 8888 is added to the global HTTP port list. Port 80 is already configured as a global HTTP port.

```
> show ns param
        Global configuration settings:
                          HTTP port(s): 80
                       Max connections: 0
           Max requests per connection: 0
                   Client IP insertion: DISABLED
                        Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
        ...
        ...
 Done
> set ns param -httpPort 80 8888
 Done
>  show ns param

        Global configuration settings:
                          HTTP port(s): 80,8888
                       Max connections: 0
           Max requests per connection: 0
                   Client IP insertion: DISABLED
                        Cookie version: 0
        Persistence Cookie Secure Flag: ENABLED
                          Min Path MTU: 576
```

```

          ...
          ...
 Done
>
<!--NeedCopy-->
```

To configure a global HTTP port by using the configuration utility

1. Navigate to **System** > **Settings** > **Change HTTP Parameters**, and then add an HTTP port number.

Set up basic load balancing

September 21, 2020

Before configuring your initial load balancing setup, enable the load balancing feature. Then begin by creating at least one service for each server in the load balancing group. With the services configured, you are ready to create a load balancing virtual server, and bind each service to the virtual server. That completes the initial setup. Before proceeding with further configuration, verify your configuration to make sure that each element was configured properly and is operating as expected.

Enabling Load Balancing

You can configure load balancing entities such as services and virtual servers when the load balancing feature is disabled, but they will not function until you enable the feature.

To enable load balancing by using the CLI

At the command prompt, type the following command to enable load balancing and verify the configuration:

- enable ns feature LB
- show ns feature

Example

```
    > enable ns feature LoadBalancing

     Done

    > show ns feature



            Feature                        Acronym              Status

            -------                        -------              ------

     1)     Web Logging                    WL                   OFF

     2)     Surge Protection               SP                   ON

     3) Load Balancing LB ON

     .

     .

     .

     24)    NetScaler Push                 push                 OFF

     Done
<!--NeedCopy-->
```

To enable load balancing by using the GUI

Navigate to **System** > **Settings** and, in **Configure Basic Features**, select **Load Balancing**.

Configuring a Server Object

Create an entry for your server on the NetScaler appliance. THe NetScaler appliance supports IP address based servers and domain-based servers. If you create an IP address based server, you can specify the name of the server instead of its IP address when you create a service. For information about setting up DNS for a domain-based server, see Domain Name System.

To create a server object by using the CLI

At the command prompt, type:

```
add server <name>@ <IPAddress>@ | <domain>
<!--NeedCopy-->
```

Example:

```
add server web_serv 10.102.27.150
<!--NeedCopy-->
```

To create a server object by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Servers**, and add a server object.

Configuring Services

After you enable the load balancing feature, you must create at least one service for each application server that is to be included in your load balancing setup. The services that you configure provide the connections between the NetScaler appliance and the load balanced servers. Each service has a name and specifies an IP address, a port, and the type of data that is served.

If you create a service without first creating a server object, the IP address of the service is also the name of the server that hosts the service. If you prefer to identify servers by name rather than IP address, you can create server objects and then specify a server's name instead of its IP address when you create a service.

When you create a service that uses UDP as the transport layer protocol, a ping monitor is automatically bound to the service. A ping monitor is the most basic of the built-in monitors. When you create a service that uses TCP as the transport layer protocol, a TCP_default monitor is automatically bound to the service. When you develop a strategy for managing your load balancing setup, you might decide to bind a different type of monitor, or multiple monitors, to the service.

Creating a Service

Before you create a service, you need to understand the different service types and how each is used. The following list describes the types of services supported on the NetScaler appliance.

HTTP

Used for load-balanced servers that accept HTTP traffic, such as standard websites and web applications. The HTTP service type enables the NetScaler appliance to provide compression, content filtering, caching, and client keep-alive support for your Layer 7 web servers. This service type also supports virtual server IP port insertion, redirect port rewriting, Web 2.0 Push, and URL redirection support.

Because HTTP is a TCP-based application protocol, you can also use the TCP service type for web servers. If you do so, however, the NetScaler appliance is able to perform only Layer 4 load balancing. It cannot provide any of the Layer 7 supports described earlier.

SSL

Used for servers that accept HTTPS traffic, such as ecommerce websites and shopping cart applications. The SSL service type enables the NetScaler appliance to encrypt and decrypt SSL traffic (perform SSL offloading) for your secure web applications. It also supports HTTP persistence, content switching, rewrite, virtual server IP port insertion, Web 2.0 Push, and URL redirection.

You can also use the SSL_BRIDGE, SSL_TCP, or TCP service types. If you do so, however, the appliance performs only Layer 4 load balancing. It cannot provide SSL offloading or any of the Layer 7 supports described above.

FTP

Used for servers that accept FTP traffic. The FTP service type enables the NetScaler appliance to support specific details of the FTP protocol.

You can also use TCP or ANY service types for FTP servers.

TCP

Used for servers that accept many different types of TCP traffic, or that accept a type of TCP traffic for which a more specific type of service is not available.

You can also use the ANY service type for these servers.

SSL_TCP

Used for servers that accept non-HTTP-based SSL traffic, to support SSL offloading.

You can also use the TCP service type for these services. If you do so, the NetScaler appliance performs both the Layer 4 load balancing and SSL offloading.

UDP

Used for servers that accept UDP traffic. You can also use the ANY service type.

SSL_BRIDGE

Used for servers that accept SSL traffic when you do not want the NetScaler appliance to perform SSL offloading. Alternatively, you can use the SSL_TCP service type.

NNTP

Used for servers that accept Network News Transfer Protocol (NNTP) traffic, typically Usenet sites.

DNS

Used for servers that accept DNS traffic, typically nameservers. With the DNS service type, the NetScaler appliance validates the packet format of each DNS request and response. It can also cache DNS responses. You can apply DNS policies to DNS services.

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance can only perform Layer 4 load balancing. It cannot provide support for DNS-specific features.

ANY

Used for servers that accept any type of TCP, UDP, or ICMP traffic. The ANY parameter is used primarily with firewall load balancing and link load balancing.

SIP-UDP

Used for servers that accept UDP-based Session Initiation Protocol (SIP) traffic. SIP initiates, manages, and terminates multimedia communications sessions, and has emerged as the standard for Internet telephony (VoIP).

You can also use the UDP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot provide support for SIP-specific features.

DNS-TCP

Used for servers that accept DNS traffic, where the NetScaler appliance acts as a proxy for TCP traffic sent to DNS servers. With services of the DNS-TCP service type, the NetScaler appliance validates the packet format of each DNS request and response and can cache DNS responses, as with the DNS service type.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance only performs Layer 4 load balancing of external DNS name servers. It cannot provide support for any DNS-specific features.

RTSP

Used for servers that accept Real Time Streaming Protocol (RTSP) traffic. RTSP provides delivery of multimedia and other streaming data. Select this type to support audio, video, and other types of streamed media.

You can also use the TCP service type for these services. If you do, however, the NetScaler appliance performs only Layer 4 load balancing. It cannot parse the RTSP stream or provide support for RTSPID persistence or RTSP NATting.

DHCPRA

Used for servers that accept DHCP traffic. The DHCPRA service type can be used to relay DHCP requests and responses between VLANs.

DIAMETER

Used for load balancing Diameter traffic among multiple Diameter servers. Diameter uses message-based load balancing.

SSL_DIAMETER

Used for load balancing Diameter traffic over SSL.

Services are designated as DISABLED until the NetScaler appliance connects to the associated load-balanced server and verifies that it is operational. At that point, the service is designated as ENABLED. Thereafter, the NetScaler appliance periodically monitors the status of the servers, and places any that fail to respond to monitoring probes (called health checks) back in the DISABLED state until they respond.

Note: You can create a range of services from a single CLI command or the same dialog box. The names in the range vary by a number used as a suffix/prefix. For example, service1, service2, and so on. From the configuration utility, you can specify a range only in the last octet of the IP address, which is the fourth in case of an IPv4 address and the eighth in case of an IPv6 address. From the command line, you can specify the range in any octet of the IP address.

To create a service by using the CLI

At the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>

add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

To create a service by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing** > **Services**.
2. In the details pane, click **Add**.
3. In the Create Service dialog box, specify values for the following parameters:
    - Service Name—name
    - Server—serverName
    - Protocol—serviceType
    - Port—port
4. Click **Create**, and then click **Close**. The service you created appears in the Services pane.

Creating a Virtual Server

After you create your services, you must create a virtual server to accept traffic for the load balanced Web sites, applications, or servers. Once load balancing is configured, users connect to the load-balanced website, application, or server through the virtual server's IP address or FQDN.

Note:

- Virtual server names prefixed with "app_" do not appear in the GUI though they are present in the ns.conf file and are displayed when you run the show command. However, virtual server names prefixed with "app" are displayed in the GUI.
- The virtual server is designated as DOWN until you bind the services that you created to it, and until the Citrix ADC appliance connects to those services and verifies that they are operational. Only then is the virtual server designated as UP.

To create a virtual server by using the CLI

At the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

To create a virtual server by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and then create a virtual server.

Binding Services to the Virtual Server

Note: A service can be bound to a maximum of 500 virtual servers.

After you have created services and a virtual server, you must bind the services to the virtual server. Usually, services are bound to virtual servers of the same type, but you can bind certain types of services to certain different types of virtual servers, as shown below.

Virtual Server Type	Service Type	Comment
HTTP	SSL	You would normally bind an SSL service to an HTTP virtual server to do encryption.
SSL	HTTP	You would normally bind an HTTP service to an SSL virtual server to do SSL offloading.
SSL_TCP	TCP	You would normally bind a TCP service to an SSL_TCP virtual server to do SSL offloading for other TCP (SSL decryption without content awareness).

The state of the services bound to a virtual server determines the state of the virtual server: if all of the bound services are DOWN, the virtual server is marked DOWN, and if any of the bound services is UP or OUT OF SERVICE, the state of the virtual server is UP.

To bind a service to a load balancing virtual server by using the CLI

At the command prompt, type:

```
bind lb vserver <name> <serviceName>

bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To bind a service to a load balancing virtual server by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing** > **Virtual Servers**, and select a virtual server.
2. Click in the **Service** section, and select a service to bind.

Note: You can bind a service to multiple virtual servers.

Verifying the Configuration

After finishing your basic configuration, you should view the properties of each service and load balancing virtual server in your load balancing setup to verify that each is configured correctly. After the configuration is up and running, you should view the statistics for each service and load balancing virtual server to check for possible problems.

Viewing the Properties of a Server Object

You can view properties such as the name, state, and IP address of any server object in your NetScaler appliance configuration.

To view the properties of server objects by using the CLI

At the command prompt, type:

```
show server <serverName>

show server server-1
<!--NeedCopy-->
```

To view the properties of server objects by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Servers**. The parameter values of the available servers appear in the details pane.

Viewing the Properties of a Virtual Server

You can view properties such as the name, state, effective state, IP address, port, protocol, method, and number of bound services for your virtual servers. If you have configured more than the basic load balancing settings, you can view the persistence settings for your virtual servers, any policies that are bound to them, and any cache redirection and content switching virtual servers that have been bound to the virtual servers.

To view the properties of a load balancing virtual server by using the CLI

At the command prompt, type:

```
show lb vserver <name>

show lb vserver Vserver-LB-1
<!--NeedCopy-->
```

To view the properties of a load balancing virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the details pane, click a virtual server to display its properties at the bottom of the details pane.
3. To view cache redirection and content switching virtual servers that are bound to this virtual server, click **Show CS/CR Bindings**.

Viewing the Properties of a Service

You can view the name, state, IP address, port, protocol, maximum client connection, maximum requests per connection, and server type of the configured services, and use this information to troubleshoot any mistake in the service configuration.

To view the properties of services by using the CLI

At the command prompt, type:

```
show service <name>

show service Service-HTTP-1
<!--NeedCopy-->
```

To view the properties of services by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Services**. The details of the available services appear on the Services pane.

Viewing the Bindings of a Service

You can view the list of virtual servers to which the service is bound. The binding information also provides the name, IP address, port, and state of the virtual servers to which the services are bound. You can use the binding information to troubleshoot any problem with binding the services to virtual servers.

To view the bindings of a service by using the command line

At the command prompt, type:

```
show service bindings <name>

show service bindings Service-HTTP-1
<!--NeedCopy-->
```

To view the bindings of a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. In the details pane, select the service whose binding information you want to view.
3. In the **Action** tab, click **Show Bindings**.

Viewing the Statistics of a Virtual Server

To evaluate the performance of virtual servers or to troubleshoot problems, you can display details of the virtual servers configured on the NetScaler appliance. You can display a summary of statistics for all the virtual servers, or you can specify the name of a virtual server to display the statistics only for that virtual server. You can display the following details:

- Name
- IP address
- Port
- Protocol
- State of the virtual server
- Rate of requests received
- Rate of hits

To display virtual server statistics by using the CLI

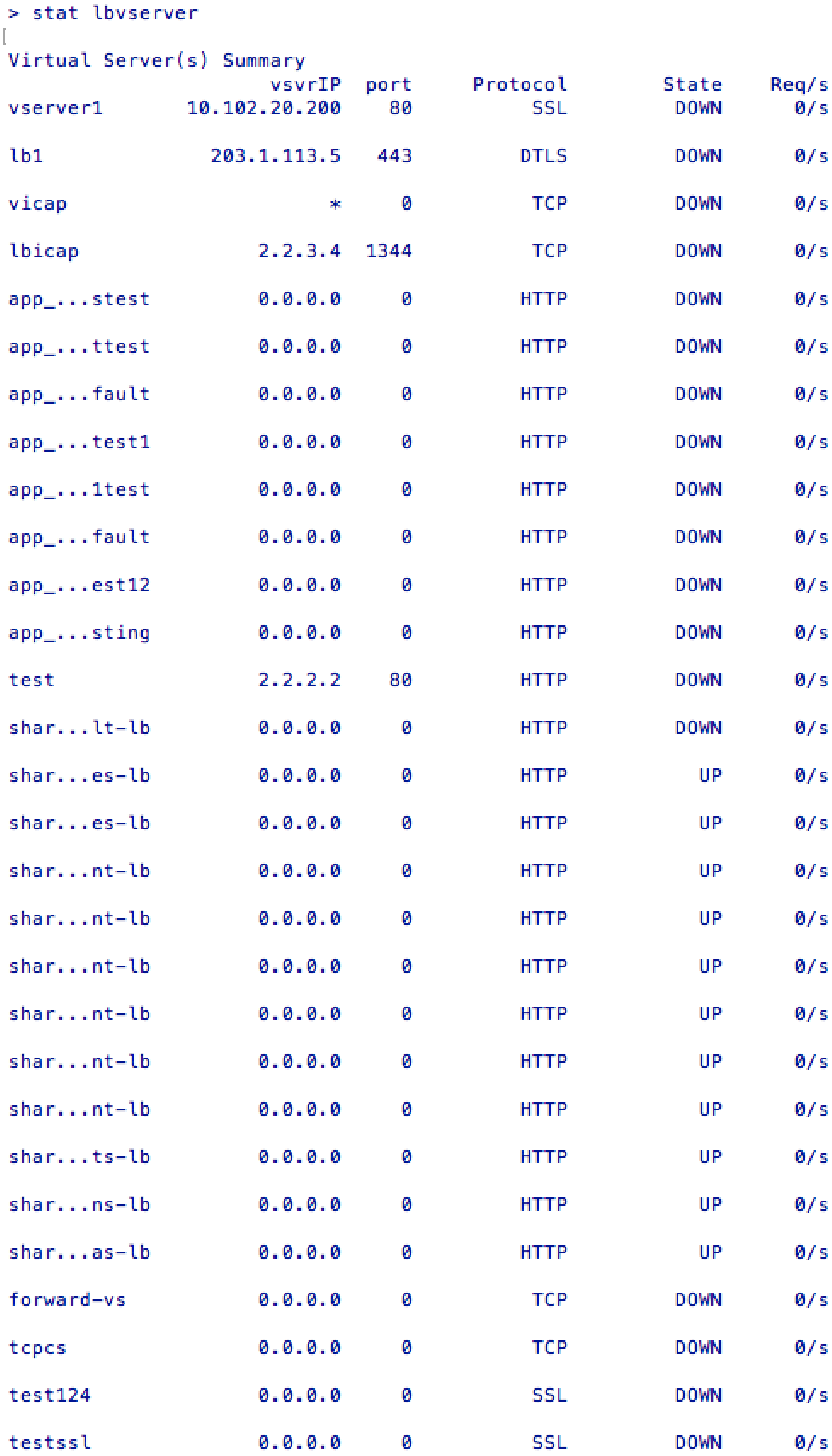
To display a summary of the statistics for all the virtual servers currently configured on the appliance, or for a single virtual server, at the command prompt, type:

stat lb vserver [<`name`>]

Example:

```
stat lb vserver server-1
<!--NeedCopy-->
```

The following figure displays a sample statistic page.



```
> stat lbvserver
[
Virtual Server(s) Summary
                      vsvrIP  port     Protocol        State    Req/s
vserver1       10.102.20.200    80          SSL         DOWN      0/s

lb1              203.1.113.5   443         DTLS         DOWN      0/s

vicap                      *     0          TCP         DOWN      0/s

lbicap               2.2.3.4  1344          TCP         DOWN      0/s

app_...stest         0.0.0.0     0         HTTP         DOWN      0/s

app_...ttest         0.0.0.0     0         HTTP         DOWN      0/s

app_...fault         0.0.0.0     0         HTTP         DOWN      0/s

app_...test1         0.0.0.0     0         HTTP         DOWN      0/s

app_...1test         0.0.0.0     0         HTTP         DOWN      0/s

app_...fault         0.0.0.0     0         HTTP         DOWN      0/s

app_...est12         0.0.0.0     0         HTTP         DOWN      0/s

app_...sting         0.0.0.0     0         HTTP         DOWN      0/s

test                 2.2.2.2    80         HTTP         DOWN      0/s

shar...lt-lb         0.0.0.0     0         HTTP         DOWN      0/s

shar...es-lb         0.0.0.0     0         HTTP           UP      0/s

shar...es-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...nt-lb         0.0.0.0     0         HTTP           UP      0/s

shar...ts-lb         0.0.0.0     0         HTTP           UP      0/s

shar...ns-lb         0.0.0.0     0         HTTP           UP      0/s

shar...as-lb         0.0.0.0     0         HTTP           UP      0/s

forward-vs           0.0.0.0     0          TCP         DOWN      0/s

tcpcs                0.0.0.0     0          TCP         DOWN      0/s

test124              0.0.0.0     0          SSL         DOWN      0/s

testssl              0.0.0.0     0          SSL         DOWN      0/s
```

To display virtual server statistics by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. If you want to display the statistics for only one virtual server, in the details pane, select the virtual server whose statistics you want to display.
3. In the details pane, click **Statistics**.

Viewing the Statistics of a Service

You can view the rate of requests, responses, request bytes, response bytes, current client connections, requests in surge queue, current server connections, and so forth using the service statistics.

To view the statistics of a service by using the CLI

At the command prompt, type:

```
stat service `<name>`
<!--NeedCopy-->
```

Example:

```
stat service Service-HTTP-1
<!--NeedCopy-->
```

To view the statistics of a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. In the details pane, select the service whose statistics you want to view (for example, Service-HTTP-1).
3. Click **Statistics**. The statistics appear in a new window.

Load balance virtual server and service states

September 21, 2020

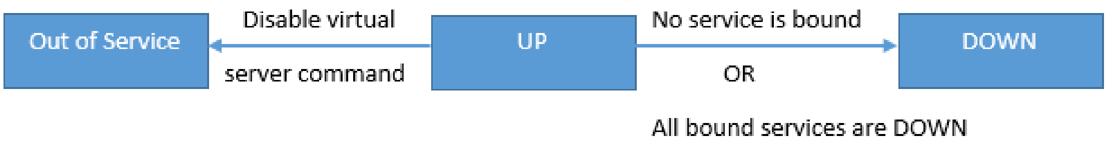
A load balancing virtual server that does not have a backup virtual server can take the following states, depending on the states of the service(s) bound to it and whether it is administratively disabled:

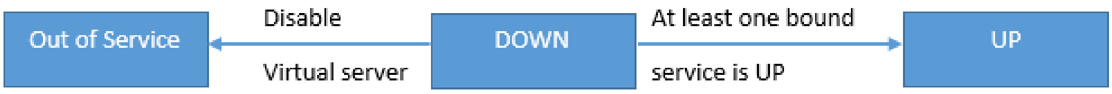
- **UP**: At least one of the services bound to the virtual server is UP.

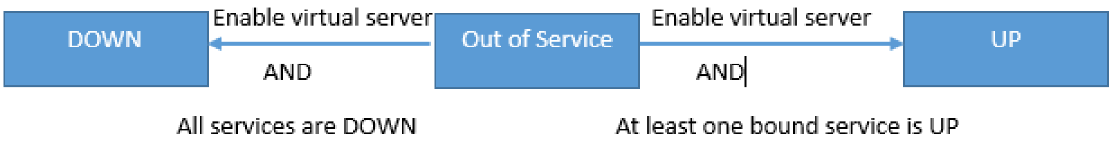
- **DOWN**: All the services bound to the virtual server are DOWN, or the load balancing feature is not enabled.
- **Out of Service (OFS)**: If you administratively disable the virtual server, it enters the OFS state but its effective state is DOWN. Transitioning to the OFS state from the DOWN or UP state, or to the DOWN or UP state from the OFS state, is controlled by the administrator.

The state and effective state of a virtual server are the same if a backup virtual server is not configured. However, if a backup virtual server or a chain of backup virtual servers is configured, the effective state is derived from the states of the services that are bound to the primary virtual server and the backup virtual server(s). If any of the backup virtual servers in the chain is UP, the effective state of the primary virtual server is UP, even if all the services bound to the primary virtual server are DOWN.

The following diagrams show the conditions under which a virtual server transitions from one state to another.


Out of Service
Disable virtual
server command
UP
No service is bound
OR
All bound services are DOWN
DOWN


Out of Service
Disable
Virtual server
DOWN
At least one bound
service is UP
UP


DOWN
Enable virtual server
AND
All services are DOWN
Out of Service
Enable virtual server
AND
At least one bound service is UP
UP

A service can take the following states:

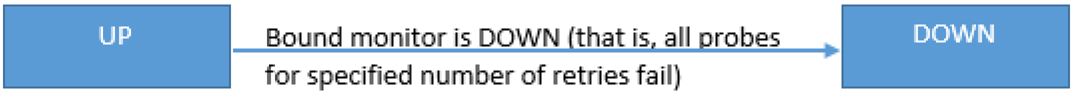
- **UP:** If probes from all the monitors bound to the service are successful.
- **DOWN**: If monitoring probes to the service are not answered within the configured time limit.
- **OUT OF SERVICE**: If you administratively disable the service, or if you gracefully shut down the service and there are no active transactions to the service

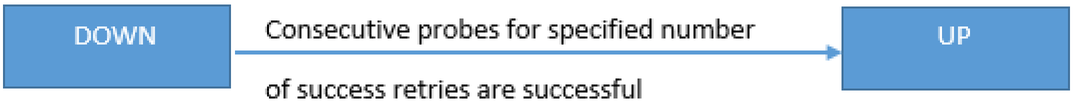
- **GOING OUT OF SERVICE (TROFS)**: If you administratively disable the service with delay, or gracefully shut down the service and there are active transactions to the service. For more information, see Graceful Shut down of Services.
- **DOWN WHEN GOING OUT OF SERVICE (TROFS_DOWN)[]** A monitoring probe fails while the service is in the GOING OUT OF SERVICE state.

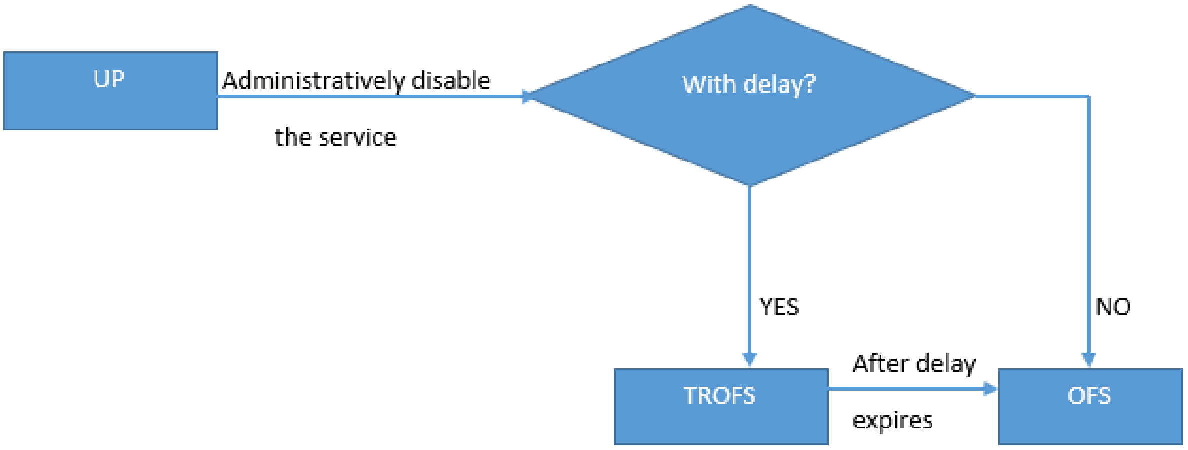
A service in the process of transitioning from UP to OFS is in the GOING OUT OF SERVICE state. A service transitioning from DOWN to OFS is in the DOWN WHEN GOING OUT OF SERVICE state. For example, if a service is DOWN and you disable it with delay, the service transitions to DOWN WHEN GOING OUT OF SERVICE and then to the OUT OF SERVICE state. If a service is UP and you disable it with delay, the service transitions to GOING OUT OF SERVICE. During this time, if a monitoring probe to the server fails, the service transitions to DOWN WHEN GOING OUT OF SERVICE and, after the delay time expires, enters the OFS state.

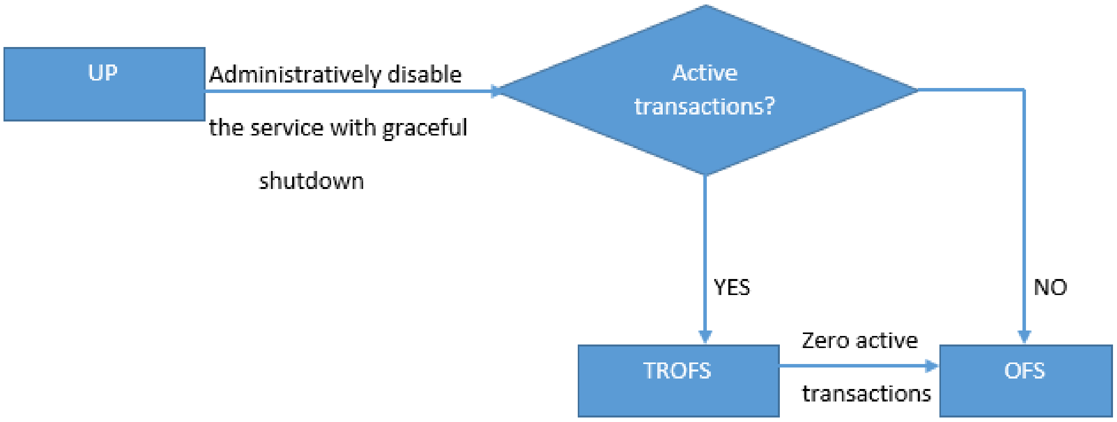
> **Note**
>
> You can configure spillover to a backup virtual server by setting the "healthThreshold" parameter to a non-zero positive value. Then, if a single service bound to the primary virtual server transitions to the DOWN WHEN GOING OUT OF SERVICE state and the health threshold is not reached, the primary virtual server is marked DOWN and new connections are directed to the backup virtual server.

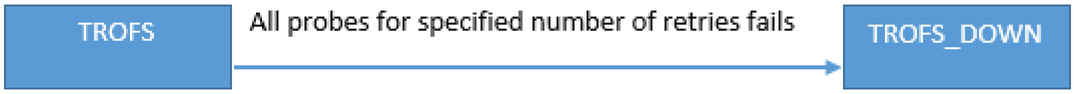
The following diagrams show the conditions under which a service transitions from one state to another.


UP
Bound monitor is DOWN (that is, all probes
for specified number of retries fail)
DOWN


DOWN
Consecutive probes for specified number
of success retries are successful
UP


UP
Administratively disable
the service
With delay?
YES
NO
TROFS
After delay
expires
OFS


UP
Administratively disable
the service with graceful
shutdown
Active
transactions?
YES
NO
TROFS
Zero active
transactions
OFS


TROFS
All probes for specified number of retries fails
TROFS_DOWN

Support for load balancing profile

September 21, 2020

A load balancing configuration has a large number of parameters, so setting the same parameters on a number of virtual servers can become tedious. From release 11.1, a load balancing (LB) profile makes this task easier. You can now set load balancing parameters in a profile and associate this profile with virtual servers, instead of setting these parameters on each virtual server.

The following parameters are presently supported in an LB profile:

- HTTPonlyflag—Include the HttpOnly attribute in persistence cookies. The HttpOnly attribute limits the scope of a cookie to HTTP requests and helps mitigate the risk of cross-site scripting attacks.
- UseSecuredPersistenceCookie—Encrypt the persistence cookie values by using a SHA2 hash algorithm.
- Cookiepassphrase—Specify the passphrase used to generate a secured persistence cookie value.
- DBS_LB—Enable database specific load balancing for MySQL and MSSQL service types.
- Cl_process_local—Packets destined to a virtual server in a cluster are not steered. Enable option for single packet request response mode or when the upstream device is performing a proper RSS for connection based distribution.

> **Note**
>
> You can set DBS_LB and Cl_process_local parameters on a virtual server and in the profile. If you enable these parameters on a virtual server and then set a profile to this virtual server, the parameters appear as disabled in the output of the "show lb vserver" command for that virtual server. Check the profile to see the actual status of these parameters. In addition, if you set and then unset a profile to a virtual server, the parameters will be set with default values for that virtual server.

To create an LB profile by using the CLI

At the command prompt, type:

```
add lb profile <lbprofilename> -dbsLb ( ENABLED | DISABLED ) -processLocal ( ENABLED | DISABLED ) -httpOnlyCookieFlag ( ENABLED | DISABLED ) -cookiePassphrase -useSecuredPersistenceCookie ( ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
add lb profile p1

Done

show lb profile p1

LB Profile name:                 p1
```

```

DBS LB : DISABLED        Process Local: DISABLED

Persistence Cookie HttpOnly Flag: ENABLED

Use Secured Persistence Cookie Flag: DISABLED

No of vservers bound: 0

Done
<!--NeedCopy-->
```

To create an LB profile by using the GUI

Navigate to **System** > **Profiles** > **LB Profile**, and add a profile.

To associate an LB profile with an LB virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -lbprofilename <string>
<!--NeedCopy-->
```

Example

```
set lbvserver lbvip1 -lbprofile p1

Done

sh lb vserver lbvip1

lbvip1 (203.0.113.1:80) - HTTP       Type: ADDRESS
State: UP
Last state change was at Wed May 25 12:36:20 2016
Time since last state change: 0 days, 00:01:26.140
Effective State: UP  ARP:DISABLED
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
Appflow logging: ENABLED
Port Rewrite : DISABLED
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN
Mode: IP
```

```
Persistence: NONE
Vserver IP and Port insertion: OFF
Push: DISABLED  Push VServer:
Push Multi Clients: NO
Push Label Rule: none
L2Conn: OFF
Skip Persistency: None
Listen Policy: NONE
IcmpResponse: PASSIVE
RHIstate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED
Traffic Domain: 0
LB Profile: p1
Done
<!--NeedCopy-->
```

To associate an LB profile with an LB virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Select a virtual server, and click **Edit**.
3. In **Advanced Settings**, click **Profiles**.
4. In the **LB Profile** list, select the profile to associate with this virtual server.

Load balancing algorithms

September 21, 2020

The load balancing algorithm defines the criteria that the NetScaler appliance uses to select the service to which to redirect each client request. Different load balancing algorithms use different criteria. For example, the least connection algorithm selects the service with the fewest active connections, while the round robin algorithm maintains a running queue of active services, distributes each connection to the next service in the queue, and then sends that service to the end of the queue.

Some load balancing algorithms are best suited to handling traffic on websites, others to managing traffic to DNS servers, and others to handling complex web applications used in e-commerce or on company LANs or WANs. The following table lists each load balancing algorithm that the NetScaler appliance supports, with a brief description of how each operates.

Name	Server selection based on
LEASTCONNECTION	Which service currently has the fewest client connections. This is the default load balancing algorithm.
ROUNDROBIN	Which service is at the top of a list of services. After that service is selected for a connection, it moves to the bottom of the list.
LEASTRESPONSETIME	Which load balanced server currently has the quickest response time.
URLHASH	A hash of the destination URL.
DOMAINHASH	A hash of the destination domain.
DESTINATIONIPHASH	A hash of the destination IP address.
SOURCEIPHASH	A hash of the source IP address.
SRCIPDESTIPHASH	A hash of the source and destination IP addresses.
CALLIDHASH	A hash of the call ID in the SIP header.
SRCIPSRCPORTHASH	A hash of the client's IP address and port.
LEASTBANDWIDTH	Which service currently has the fewest bandwidth constraints.
LEASTPACKETS	Which service currently is receiving the fewest packets.
CUSTOMLOAD	Data from a load monitor.
TOKEN	The configured token.
LRTM	Fewest active connections and the lowest average response time.

Depending on the protocol of the service that it is load balancing, the NetScaler appliance sets up each connection between client and server to last for a different time interval. This is called load balancing granularity, of which are three types: request-based, connection-based, and time-based granularity. The following table describes each type of granularity and when each is used.

Granularity	Types of Load Balanced Service	Specifies
Request -based	HTTP or HTTPS	A new service is chosen for each HTTP request, independent of TCP connections. As with all HTTP requests, after the Web server fulfills the request, the connection is closed.
Connection-based	TCP and TCP-based protocols other than HTTP	A service is chosen for every new TCP connection. The connection persists until terminated by either the service or the client.
Time-based	UDP and other IP protocols	A new service is chosen for each UDP packet. Upon selection of a service, a session is created between the service and a client for a specified period of time. When the time expires, the session is deleted and a new service is chosen for any additional packets, even if those packets come from the same client.

During startup of a virtual server, or whenever the state of a virtual server changes, the virtual server can initially use the round robin method to distribute the client requests among the physical servers. This type of distribution, referred to as *startup round robin,* helps prevent unnecessary load on a single server as the initial requests are served. After using the round robin method at the startup, the virtual server switches to the load balancing method specified on the virtual server.

The Startup RR Factor works in the following manner:

- If the Startup RR Factor is set to zero, the appliance switches to the specified load balancing method depending on the request rate.
- If the Startup RR Factor is any number other than zero, the appliance uses the round robin method for the specified number of requests before switching to the specified load balancing

method.
- By default, the Startup RR Factor is set to zero.

Note: You cannot set the startup RR Factor for an individual virtual server. The value you specify applies to all the virtual servers on the NetScaler appliance.

To set the startup round-robin factor by using the CLI

At the command prompt, type:

set lb parameter -startupRRFactor

Example

set lb parameter -startupRRFactor 25000

To set the startup round-robin factor by using the GUI

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters, and set the Startup RR Factor.

Least connection method

September 21, 2020

When a virtual server is configured to use the least connection load balancing algorithm (or method), it selects the service with the fewest active connections. This is the default method, because, in most circumstances, it provides the best performance.

For TCP, HTTP, HTTPS, and SSL_TCP services, the NetScaler appliance includes the following connection types in its list of existing connections:

- **Active connections to a service.** Connections representing requests that a client has sent to the virtual server and that the virtual server has forwarded to a service. For HTTP and HTTPS services, active connections represent only those HTTP or HTTPS requests that have not yet received a response.
- **Waiting connections in the surge queue.** Any connections to the virtual server that are waiting in a surge queue and have not yet been forwarded to a service. Connections can build up in the surge queue at any time, for any of the following reasons:
  - Your services have connection limits, and all services in your load balancing configuration are at that limit.
  - The surge protection feature is configured and has been activated by a surge in requests to the virtual server.

  - The load-balanced server has reached an internal limit and therefore does not open any new connections. (For example, an Apache server's connection limit is reached.)

When a virtual server uses the least connection method, it considers the waiting connections as belonging to the specific service. Therefore, it does not open new connections to those services.

For UDP services, the connections that the least connection algorithm considers include all sessions between the client and a service. These sessions are logical, time-based entities. When the first UDP packet in a session arrives, the NetScaler appliance creates a session between the source IP address and port and the destination IP address and port.

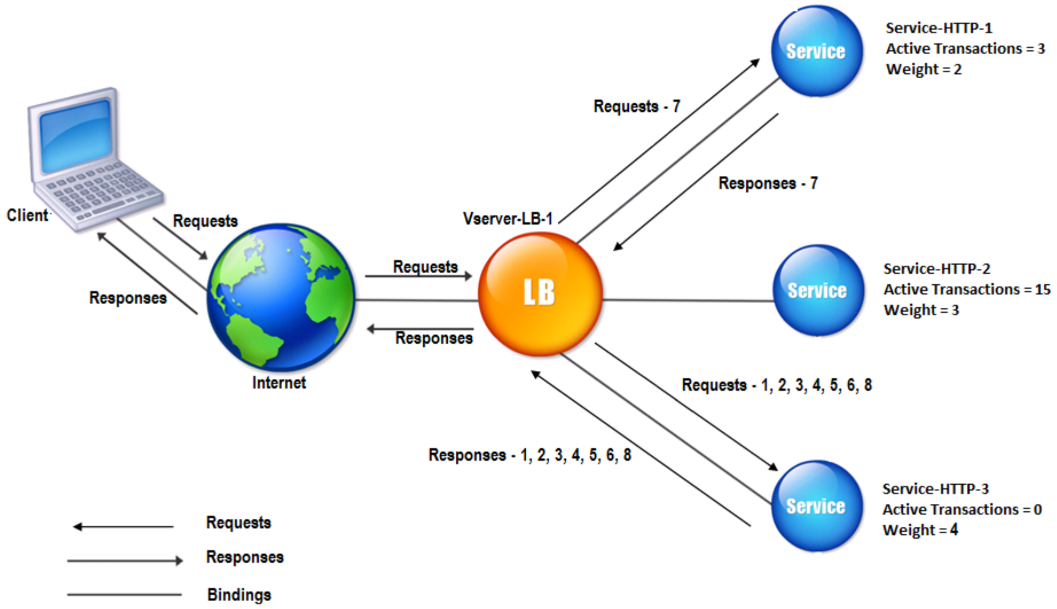
For Real-Time Streaming Protocol (RTSP) connections, the NetScaler appliance uses the number of active control connections to determine the lowest number of connections to an RTSP service.

The following example shows how a virtual server selects a service for load balancing by using the least connection method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions.
- Service-HTTP-2 is handling 15 active transactions.
- Service-HTTP-3 is not handling any active transactions.

The following diagram illustrates how the NetScaler appliance forwards incoming requests when using the least connection method.

Figure 1. Mechanism of the Least Connections Load Balancing Method


Service
Service-HTTP-1
Active Transactions = 3
Weight = 2
Requests - 7
Responses - 7
Client
Requests
Responses
Vserver-LB-1
Requests
LB
Responses
Internet
Service
Service-HTTP-2
Active Transactions = 15
Weight = 3
Requests - 1, 2, 3, 4, 5, 6, 8
Responses - 1, 2, 3, 4, 5, 6, 8
Service
Service-HTTP-3
Active Transactions = 0
Weight = 4
Requests
Responses
Bindings

In this diagram, the virtual server selects the service for each incoming connection by choosing the server with the fewest active transactions.

Connections are forwarded as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  Note: The service with no active transaction is selected first.

- Service-HTTP-3 receives the second and third requests because the service has the next least number of active transactions.
- Service-HTTP-1 receives the fourth request Because Service-HTTP-1 and Service-HTTP-3 have same number of active transactions, the virtual server uses the round robin method to choose between them.
- Service-HTTP-3 receives the fifth request.
- Service-HTTP-1 receives the sixth request, and so on, until both Service-HTTP-1 and Service-HTTP-3 are handling the same number of requests as Service-HTTP-2. At that time, the NetScaler appliance starts forwarding requests to Service-HTTP-2 when it is the least loaded service or its turn comes up in the round robin queue.

  Note: If connections to Service-HTTP-2 close, it might get new connections before each of the other two services has 15 active transactions.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
Request-1	Service-HTTP-3; (N = 0)	1	Service-HTTP-3 has the fewest active connections.
Request-2	Service-HTTP-3; (N = 1)	2	Service-HTTP-3 has the fewest active connections.
Request-3	Service-HTTP-3; (N = 2)	3	-
Request-4	Service-HTTP-1; (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.

Incoming Connection	Service Selected	Current Number of Active Connections	Remarks
Request-5	Service-HTTP-3; (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-6	Service-HTTP-1;(N = 4)	5	-
Request-7	Service-HTTP-3; (N = 4)	5	-
Request-8	Service-HTTP-1; (N = 5)	6	-

Service-HTTP-2 is selected for load balancing when it completes its active transactions and the current connections to it close, or when the other services (Service-HTTP-1 and Service-HTTP-3) have 15 or more connections each.

The NetScaler appliance can also use the least connection method when weights are assigned to services. It selects a service by using the value (Nw) of the following expression:

Nw = (Number of active transactions) * (10000 / weight)

The following example shows how the NetScaler appliance selects a service for load balancing by using the least connection method when weights are assigned to services. In the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. Connections are forwarded as follows:

- Service-HTTP-3 receives the first because the service is not handling any active transactions.

  Note: If services are not handling any active transactions, the NetScaler appliance uses the round robin method regardless of the weights assigned to each of the services.

- Service-HTTP-3 receives the second, third, fourth, fifth, sixth, and seventh requests because the service has lowest Nw value.
- Service-HTTP-1 receives the eighth request. Because Service-HTTP-1 and Service-HTTP-3 now have same Nw value, the appliance performs load balancing in a round robin manner. Therefore, Service-HTTP-3 receives the ninth request.

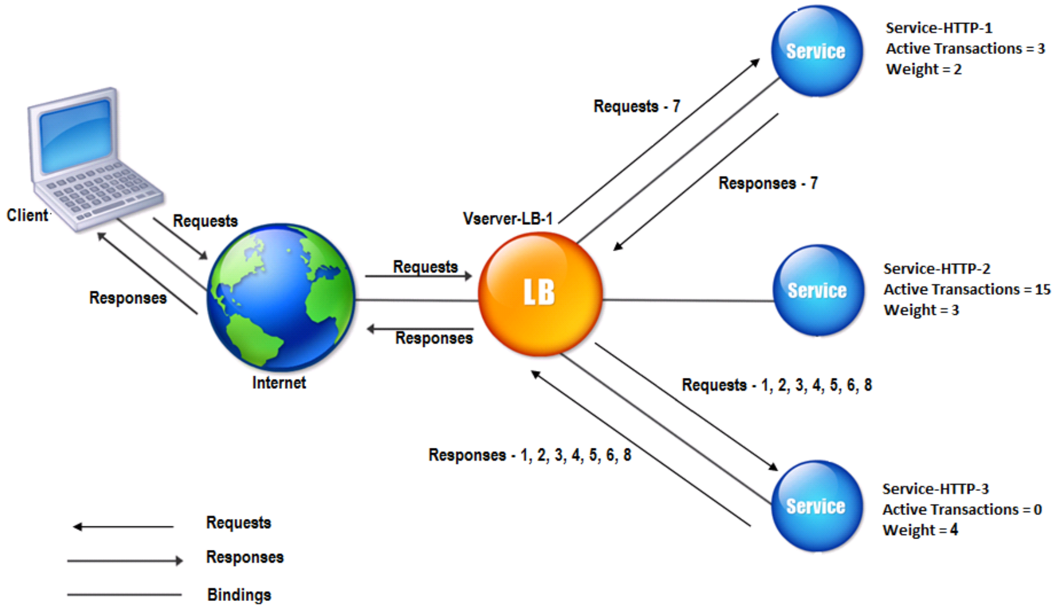
The following table explains how connections are distributed on the three-service load balancing setup that is described above.

Request Received	Service Selected	Current Nw (Number of active transactions) * (10000 / weight) value	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 2500	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 2500)	Nw = 5000	
Request-3	Service-HTTP-3; (Nw = 5000)	Nw = 7500	
Request-4	Service-HTTP-3; (Nw = 7500)	Nw = 10000	
Request-5	Service-HTTP-3; (Nw = 10000)	Nw = 12500	
Request-6	Service-HTTP-3; (Nw = 12500)	Nw = 15000	
Request-7	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw values
Request-8	Service-HTTP-3; (Nw = 15000)	Nw = 17500	

Service-HTTP-2 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-1 and Service-HTTP-3) is equal to 50000.

The following diagram illustrates how the NetScaler appliance uses the least connection method when weights are assigned to the services.

Figure 2. Mechanism of the Least Connections Load Balancing Method when Weights are Assigned


Service
Service-HTTP-1
Active Transactions = 3
Weight = 2
Requests - 7
Responses - 7
Client
Requests
Responses
Vserver-LB-1
Requests
LB
Responses
Internet
Service
Service-HTTP-2
Active Transactions = 15
Weight = 3
Requests - 1, 2, 3, 4, 5, 6, 8
Responses - 1, 2, 3, 4, 5, 6, 8
Service
Service-HTTP-3
Active Transactions = 0
Weight = 4
Requests
Responses
Bindings

To configure the least connection method, see Configuring a Load Balancing Method that Does Not Include a Policy.

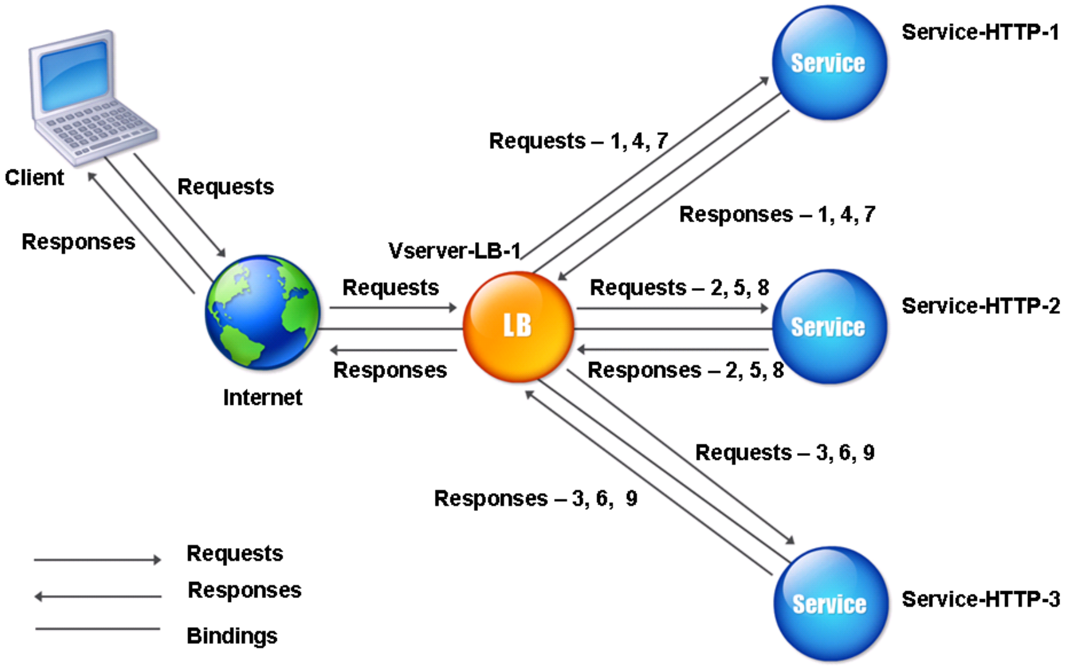
Round robin method

September 21, 2020

When a load balancing virtual server is configured to use the round robin method, it continuously rotates a list of the services that are bound to it. When the virtual server receives a request, it assigns the connection to the first service in the list, and then moves that service to the bottom of the list.

The following diagram illustrates how the NetScaler appliance uses the round robin method with a load balancing setup that contains three load-balanced servers and their associated services.

Figure 1. How the Round Robin Load Balancing Method Works


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Requests – 1, 4, 7
Responses – 1, 4, 7
Requests – 2, 5, 8
Responses – 2, 5, 8
Requests – 3, 6, 9
Responses – 3, 6, 9
Service
Service-HTTP-1
Service
Service-HTTP-2
Service
Service-HTTP-3
Requests
Responses
Bindings

If you assign a different weight to each service, the NetScaler appliance performs weighted round robin distribution of incoming connections. It does this by skipping the lower-weighted services at appropriate intervals.

For example, assume that you have a load balancing setup with three services. You set Service-HTTP-1 to a weight of 2, Service-HTTP-2 to a weight of 3, and Service-HTTP-3 to a weight of 4. The services are bound to Vserver-LB-1, which is configured to use the round robin method. With this setup, incoming requests are delivered as follows:

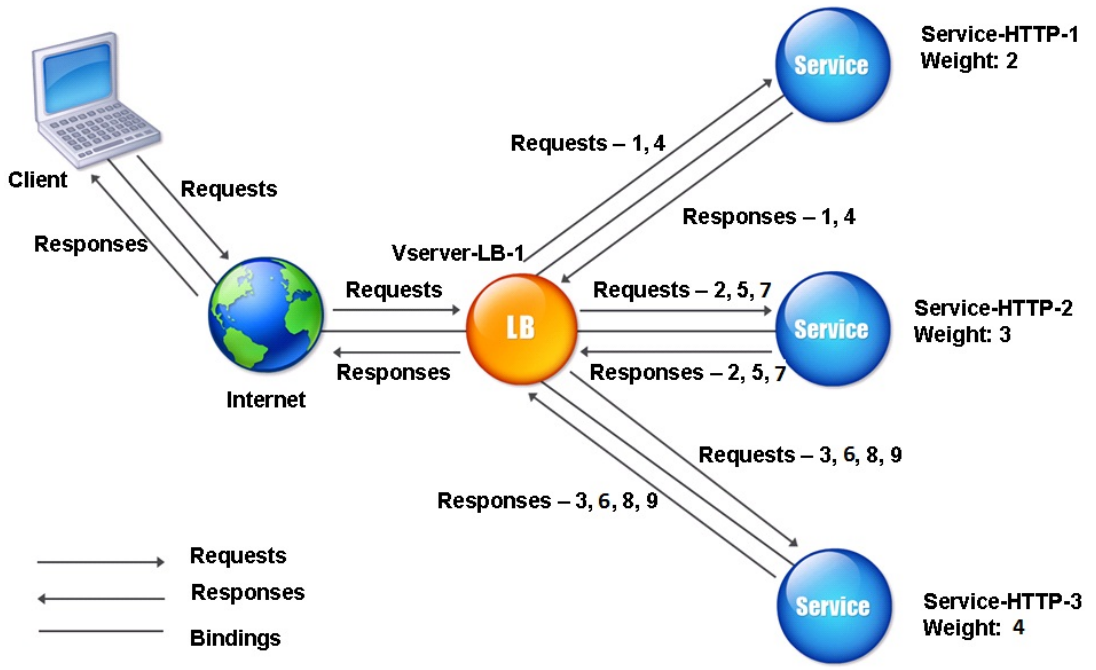
- Service-HTTP-1 receives the first request.
- Service-HTTP-2 receives the second request.
- Service-HTTP-3 receives the third request.
- Service-HTTP-1 receives the fourth request.
- Service-HTTP-2 receives the fifth request.
- Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh request.
- Service-HTTP-3 receives both the eighth and the ninth requests.

Note: You can also configure weights on services to prevent multiple services from using the same server and overloading the server.

A new cycle then begins, using the same pattern.

The following diagram illustrates the weighted round robin method.

Figure 2. How the Round Robin Load Balancing Method Works with Weighted Services


Service-HTTP-1
Weight: 2
Service
Requests – 1, 4
Client
Requests
Responses – 1, 4
Responses
Vserver-LB-1
Requests
Requests – 2, 5, 7
LB
Service
Service-HTTP-2
Weight: 3
Responses
Responses – 2, 5, 7
Internet
Requests – 3, 6, 8, 9
Responses – 3, 6, 8, 9
Requests
Responses
Bindings
Service
Service-HTTP-3
Weight: 4

To configure the round robin method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least response time method

September 21, 2020

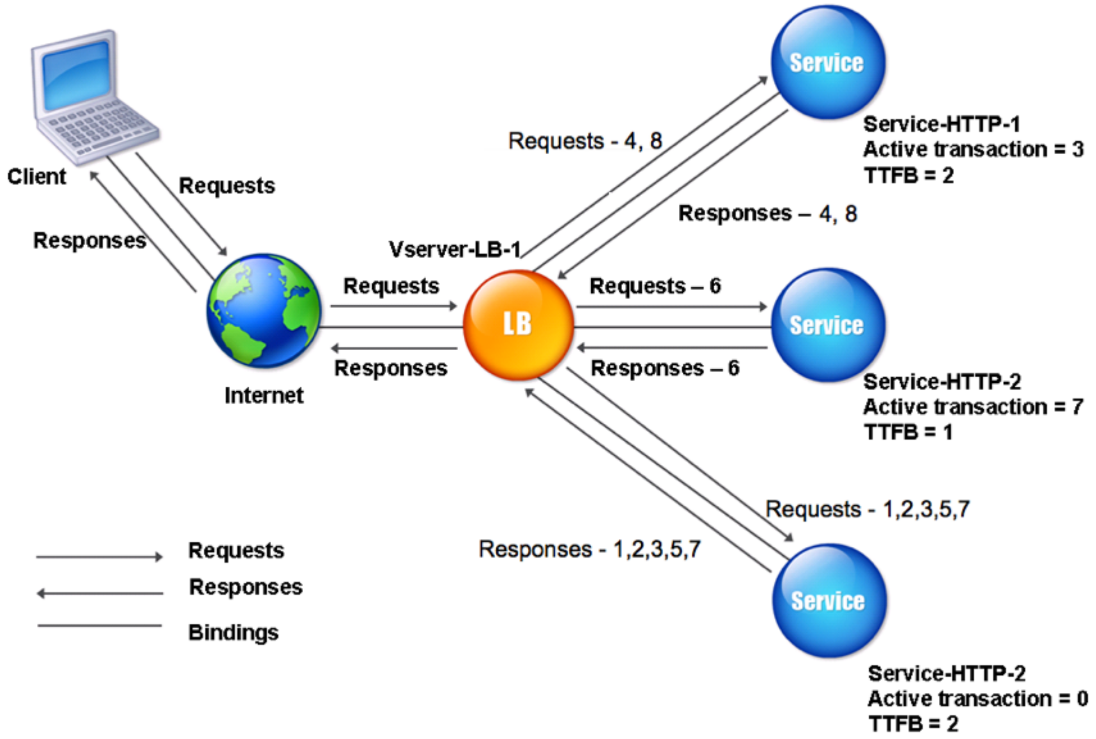
When the load balancing virtual server is configured to use the least response time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) load balancing virtual servers only. The response time (also called Time to First Byte, or TTFB) is the time interval between sending a request packet to a service and receiving the first response packet from the service. The NetScaler appliance uses response code 200 to calculate TTFB.

The following example shows how a virtual server selects a service for load balancing by using the least response time method. Consider the following three services:

- Service-HTTP-1 is handling three active transactions and TTFB is two seconds.
- Service-HTTP-2 is handling seven active transactions and TTFB is one second.
- Service-HTTP-3 is not handling any active transactions and TTFB is two seconds.

The following diagram illustrates how the NetScaler appliance uses the least response time method to forward the connections.

Figure 1. How the Least Response Time Load Balancing Method Works


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Requests - 4, 8
Responses – 4, 8
Requests – 6
Responses – 6
Requests - 1,2,3,5,7
Responses - 1,2,3,5,7
Service
Service-HTTP-1
Active transaction = 3
TTFB = 2
Service
Service-HTTP-2
Active transaction = 7
TTFB = 1
Service
Service-HTTP-2
Active transaction = 0
TTFB = 2
Requests
Responses
Bindings

The virtual server selects a service by multiplying the number of active transactions by the TTFB for each service and then selecting the service with the lowest result. For the example shown above, the virtual server forwards requests as follows:

- Service-HTTP-3 receives the first request, because the service is not handling any active transactions.
- Service-HTTP-3 also receives the second and third requests, because the result is lowest of the three services.
- Service-HTTP-1 receives the fourth request. Since Service-HTTP-1 and Service-HTTP-3 have the same result, the NetScaler appliance chooses between them by applying the Round Robin

method.

- Service-HTTP-3 receives the fifth request.
- Service-HTTP-2 receives the sixth request, because at this point it has the lowest result.
- Because Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all have the same result at this point, the appliance switches to the round robin method, and continues to distribute connections using that method.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 has the lowest N value.
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 has the lowest N value.
Request-4	Service-HTTP-1; (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values. The appliance uses the round robin method to distribute the requests.
Request-5	Service-HTTP-3; (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-6	Service-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.

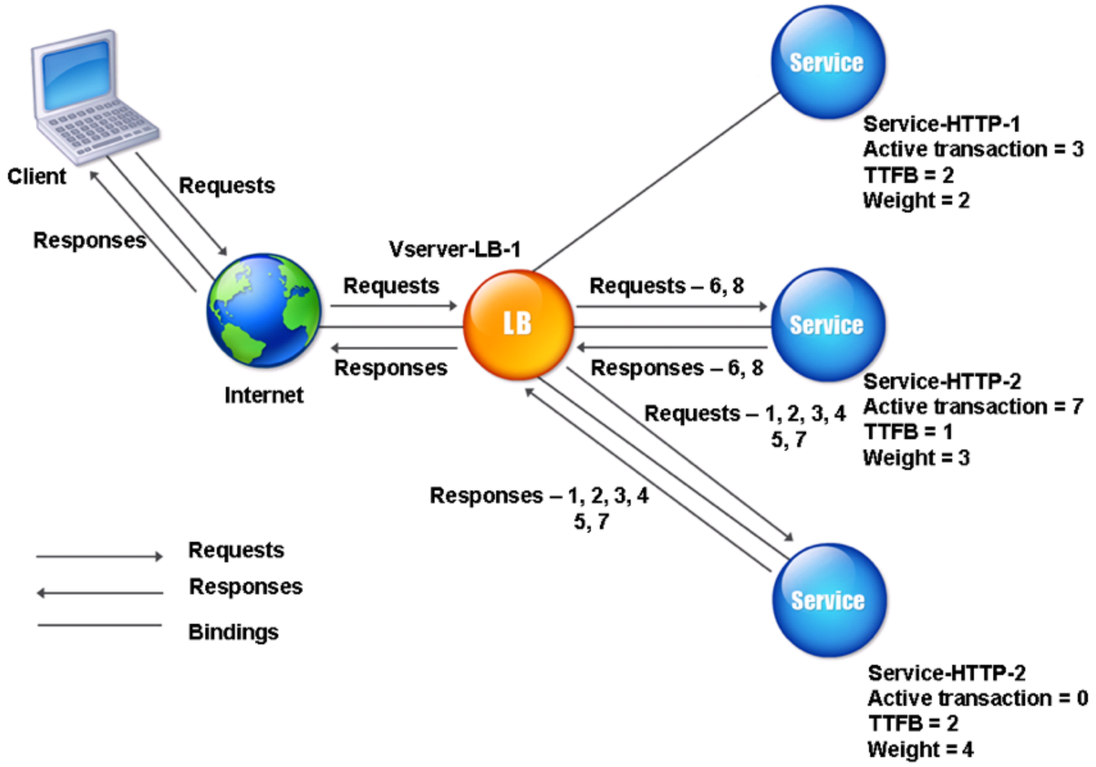
Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-7	Service-HTTP-3; (N = 8)	N = 10	Service-HTTP-1,Service-HTTP-2 and Service-HTTP-3 have the same N values. The NetScaler appliance uses the round robin method to distribute the requests.
Request-8	Service-HTTP-1; (N = 8)	N = 10	Service-HTTP-1 and Service-HTTP-2 have the same N values, the appliance uses the round robin method to distribute the requests.

Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when its N value is less than the other services (Service-HTTP-2 and Service-HTTP-3).

Selection of services when weights are assigned

The following diagram illustrates how the NetScaler appliance uses the least response time method when weights are assigned.

Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Requests – 6, 8
Responses – 6, 8
Requests – 1, 2, 3, 4 5, 7
Responses – 1, 2, 3, 4 5, 7
Service
Service-HTTP-1
Active transaction = 3
TTFB = 2
Weight = 2
Service
Service-HTTP-2
Active transaction = 7
TTFB = 1
Weight = 3
Service
Service-HTTP-2
Active transaction = 0
TTFB = 2
Weight = 4
Requests
Responses
Bindings

The virtual server selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)
Where N = (number of active transactions * TTFB)

Suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned weight of 3, and Service-HTTP-3 is assigned weight of 4.

The NetScaler appliance distributes requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.

  If services are not handling any active transactions, the appliance selects them regardless of the weights assigned to them.

- Service-HTTP-3 receives the second, third, fourth, and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and therefore the highest Nw value, so the virtual server does not select it for load balancing.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current Nw Value = (N) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 has the lowest Nw value.
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 has the lowest Nw value.
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 has the lowest Nw value.
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 has the lowest Nw value.
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when its Nw value is less than other services (Service-HTTP-2 and Service-HTTP-3).

To configure the least response time load balancing method by using the CLI

At the command prompt type;

```
set lb vserver <name> -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LEASTRESPONSETIME
<!--NeedCopy-->
```

To configure the least response time load balancing method by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **LEASTRESPONSETIME**.

For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

LRTM method

September 21, 2020

Note: LRTM stands for Least response time method using monitors (LRTM).

When a load balancing virtual server is configured to use the LRTM method, it uses the existing monitoring infrastructure to get the fastest response time. The load balancing virtual server then selects the service with the smallest number of active transactions and lowest response time. Before you use the LRTM method, you must bind application-specific monitors to each service and enable LRTM mode on these monitors. The NetScaler appliance then makes load balancing decisions based on the response times it calculates from monitoring probes.
You can use the LRTM method to load balance non-HTTP and non-HTTPS services also. You can also use this method when several monitors are bound to a service. Each monitor determines the response time by using the protocol that it measures for the service that it is bound to. The virtual server then calculates an average response time for that service by averaging the results.

The following table summarizes how response times are calculated for various monitors.

Monitor	Response time calculation
PING	Time difference between the ICMP ECHO request and the ICMP ECHO response.
TCP	Time difference between the SYN request and the SYN+ACK response.
HTTP	Time difference between the HTTP request (after the TCP connection is established) and the HTTP response.
TCP-ECV	Time difference between the time the data send string is sent and the data receive string is returned. A tcp-ecv monitor without the send and receive strings is considered to have an incorrect configuration.

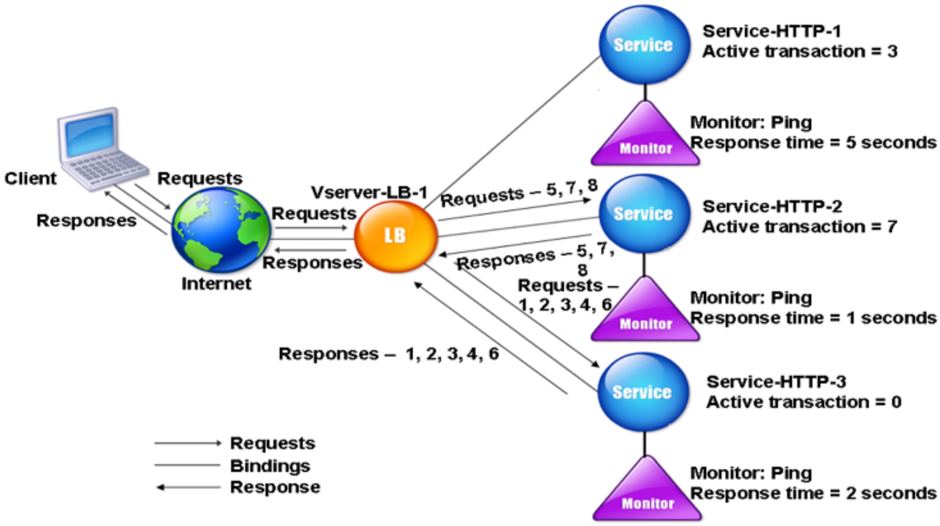
Monitor	Response time calculation
HTTP-ECV	Time difference between the HTTP request and the HTTP response.
UDP-ECV	Time difference between the UDP send string and the UDP receive string. A UDP-ECV monitor without the receive string is considered to have an incorrect configuration.
DNS	Time difference between a DNS query and the DNS response.
TCPS	Time difference between a SYN request and the SSL handshake completion.
FTP	Time difference between the sending of the user name and the completion of user authentication.
HTTPS (monitors HTTPS requests)	Time difference is same as for the HTTP monitor.
HTTPS-ECV (monitors HTTPS requests)	Time difference is same as for the HTTP-ECV monitor
USER	Time difference between the time when a request is sent to the dispatcher and the time when the dispatcher response is received.

The following example shows how the NetScaler appliance selects a service for load balancing by using the LRTM method. Consider the following three services:

- Service-HTTP-1 is handling 3 active transactions and the response time is five seconds.
- Service-HTTP-2 is handling 7 active transactions and the response time is one second.
- Service-HTTP-3 is not handling any active transactions and the response time is two seconds.

The following diagram illustrates the process that the NetScaler appliance follows when it forwards requests.

Figure 1. How the LRTM Method Works


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Requests – 5, 7, 8
Responses – 5, 7, 8
Requests – 1, 2, 3, 4, 6
Responses – 1, 2, 3, 4, 6
Service
Service-HTTP-1
Active transaction = 3
Monitor
Monitor: Ping
Response time = 5 seconds
Service
Service-HTTP-2
Active transaction = 7
Monitor
Monitor: Ping
Response time = 1 seconds
Service
Service-HTTP-3
Active transaction = 0
Monitor
Monitor: Ping
Response time = 2 seconds
Requests
Bindings
Response

The virtual server selects a service by using the value (N) in the following expression:

N = (Number of active transactions * Response time that is determined by the monitor)

The virtual server delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service is not handling any active transaction.
- Service-HTTP-3 receives the second, third and fourth requests, because this service has the lowest N value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest N value.
- Since both Service-HTTP-2 and Service-HTTP-3 currently have the same N value, the NetScaler appliance switches to the round robin method Therefore, Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the seventh and eighth requests, because this service has the lowest N value.

Service-HTTP-1 is not considered for load balancing, because it is more heavily loaded (has the highest N value) when compared to the other two services. However, if Service-HTTP-1 completes its active transactions, the NetScaler appliance again considers that service for load balancing.

The following table summarizes how N is calculated for the services.

Request Received	Service Selected	Current N Value (Number of Active Transactions * TTFB)	Remarks
Request-1	Service-HTTP-3;(N = 0)	N = 2	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-3; (N = 2)	N = 4	Service-HTTP-3 has the lowest N value.
Request-3	Service-HTTP-3; (N = 4)	N = 6	Service-HTTP-3 has the lowest N value.
Request-4	Service-HTTP-3; (N = 6)	N = 8	Service-HTTP-3 has the lowest N value.
Request-5	Service-HTTP-2; (N = 7)	N = 8	Service-HTTP-2 has the lowest N value..
Request-6	Service-HTTP-3; (N = 8)	N = 10	Service-HTTP-2 and Service-HTTP-3 have the same N values. NetScaler appliance switches to the round robin method and selects Service-HTTP-3
Request-7	Service-HTTP-2; (N = 8)	N = 9	Service-HTTP-2 has the lowest N value.
Request-8	Service-HTTP-2; (N = 9)	N = 10	Service-HTTP-2 has the lowest N value.

Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when its N value is less than the other services (Service-HTTP-2 and Service-HTTP-3).

Selection of services when weights are assigned

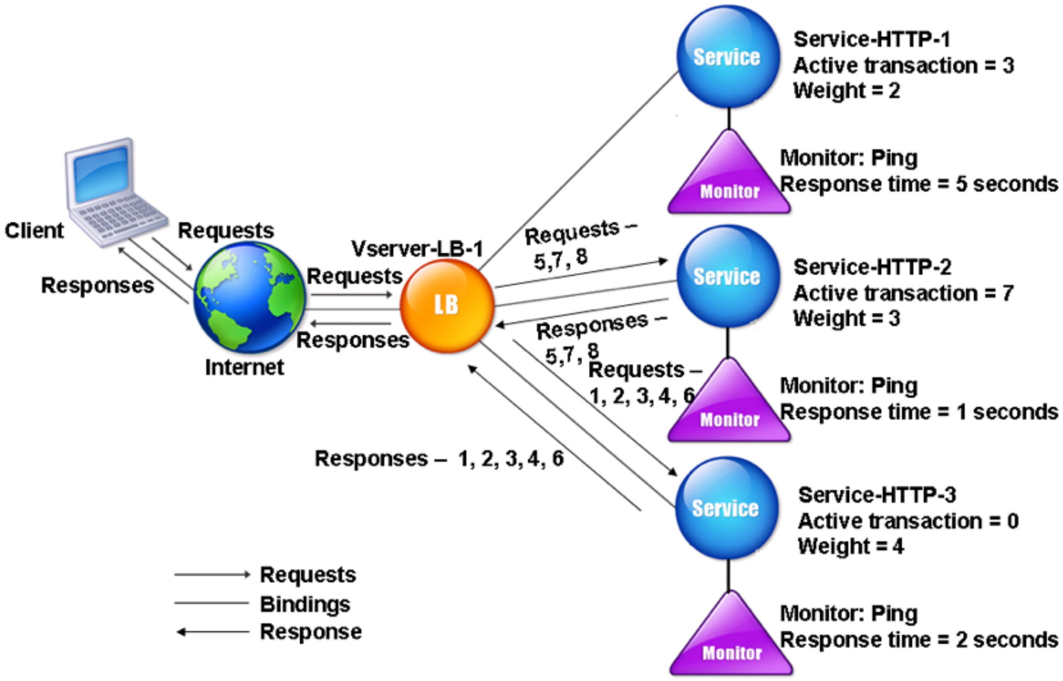
The NetScaler appliance also performs load balancing by using the number of active transactions, response time, and weights if different weights are assigned to services. The NetScaler appliance selects the service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

Where N = (Number of active transactions * Response time that is determined by the monitor)

The following diagram illustrates how the virtual server uses the LRTM method when weights are assigned.

Figure 2. How the Least Response Time Load Balancing Method Works When Weights Are Assigned


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Requests – 5,7, 8
Responses – 5,7, 8
Requests – 1, 2, 3, 4, 6
Responses – 1, 2, 3, 4, 6
Service
Service-HTTP-1
Active transaction = 3
Weight = 2
Monitor
Monitor: Ping
Response time = 5 seconds
Service
Service-HTTP-2
Active transaction = 7
Weight = 3
Monitor
Monitor: Ping
Response time = 1 seconds
Service
Service-HTTP-3
Active transaction = 0
Weight = 4
Monitor
Monitor: Ping
Response time = 2 seconds
Requests
Bindings
Response

In this example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4.
The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because it is not handling any active transactions.
- Service-HTTP-3 receives the second, third, fourth and fifth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth requests, because this service has the lowest Nw value.

Service-HTTP-1 has the lowest weight and the highest Nw value, so the NetScaler appliance does not select it for load balancing.

The following table summarizes how Nw is calculated for various monitors.

Request Received	Service Selected	Current Nw Value (N) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 0)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000	Nw = 10000	Service-HTTP-3 has the lowest Nw value.
Request-3	Service-HTTP-3; (Nw = 10000)	Nw = 15000	Service-HTTP-3 has the lowest Nw value.
Request-4	Service-HTTP-3; (Nw = 15000)	Nw = 20000	Service-HTTP-3 has the lowest Nw value.
Request-5	Service-HTTP-3; (Nw = 20000)	Nw = 25000	Service-HTTP-3 has the lowest Nw value.
Request-6	Service-HTTP-2; (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-7	Service-HTTP-3; (Nw = 25000)	Nw= 30000	Service-HTTP-3 has the lowest Nw value.
Request-8	Service-HTTP-2; (Nw = 26666.67)	Nw = 30000	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when its Nw value is less than other services (Service-HTTP-2 and Service-HTTP-3).

To configure the LRTM load balancing method by using the CLI

At the command prompt type;

```
set lb vserver <name> [-lbMethod <lbMethod>]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LRTM
<!--NeedCopy-->
```

To configure the LRTM load balancing method by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **LRTM**.

To enable the LRTM option in monitors by using the CLI

At the command prompt type;

```
set lb monitor <monitorName> <type> [-LRTM ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

Example:

```
set lb monitor monitor-HTTP-1 HTTP -LRTM ENABLED
<!--NeedCopy-->
```

To enable the LRTM option in monitors by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**, and open a monitor.
2. In Advanced Parameters, select **LRTM (Least Response Time using Monitoring)**.

For more information about configuring monitors, see Configuring Monitors in a Load Balancing Setup.

Hashing methods

September 21, 2020

Load balancing methods based on hashes of certain connection information or header information constitute the majority of the NetScaler appliance's load balancing methods. Hashes are shorter and easier to use than the information that they are based on, while retaining enough information to ensure that no two different pieces of information generate the same hash and are therefore confused with one another.

You can use the hashing load balancing methods in an environment where a cache serves a wide range of content from the Internet or specified origin servers. Caching requests reduces request and response latency, and ensures better resource (CPU) utilization, making caching popular on heavily used Web sites and application servers. Since these sites also benefit from load balancing, hashing load balancing methods are widely useful.

The NetScaler appliance provides the following hashing methods:

- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method

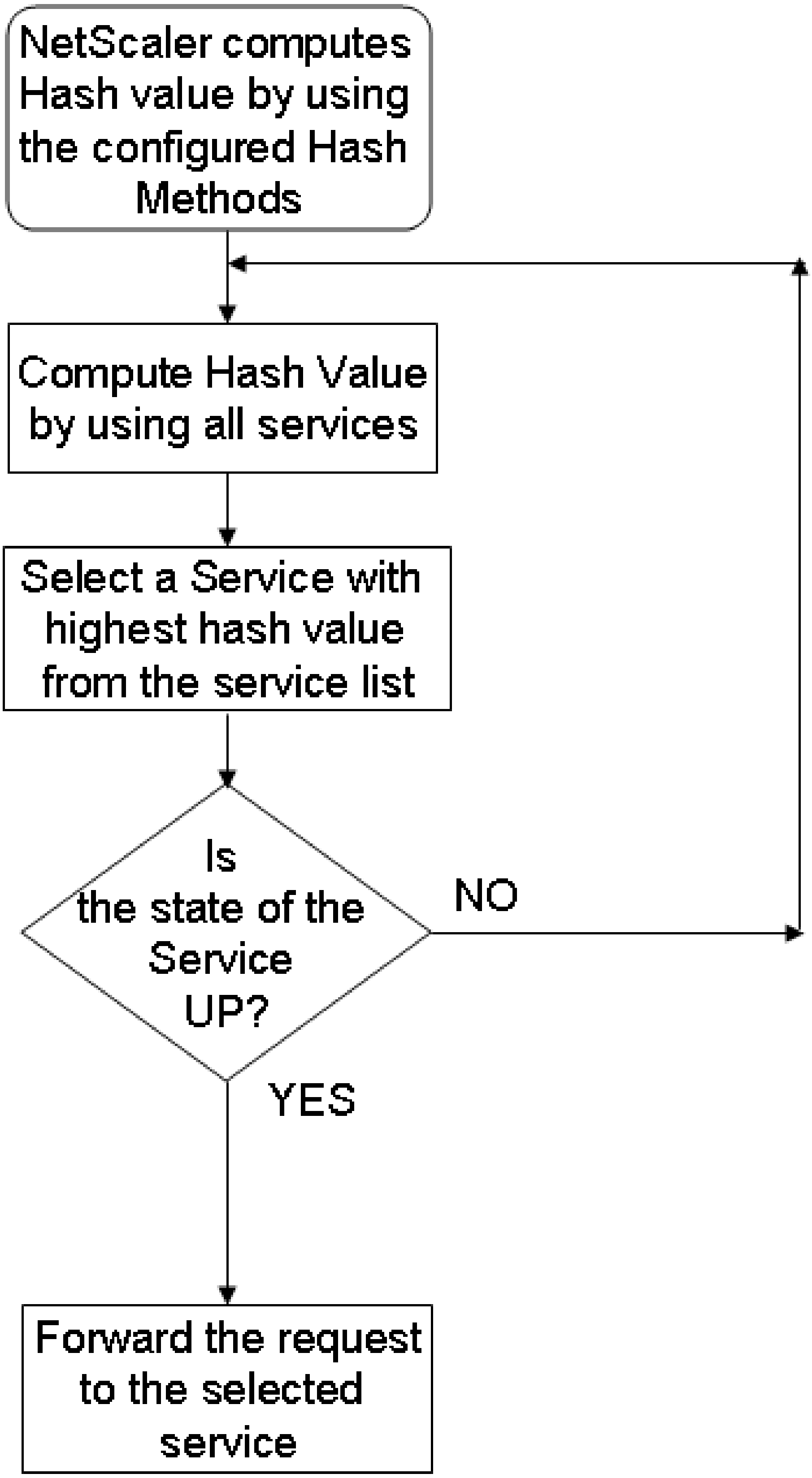
- Source IP Source Port hash method
- Call ID hash method
- Token method

These hashing algorithms ensure minimal disruption when services are added to or deleted from your load balancing setup. Most of them calculate two hash values:

- A hash of the service's IP address and port.
- A hash of the incoming URL, the domain name, the source IP address, the destination IP address, or the source and destination IP addresses, depending on the configured hash method.

The NetScaler appliance then generates a new hash value by using both of those hash values. Finally, it forwards the request to the service with highest hash value. As the appliance computes a hash value for each request and selects the service that will process the request, it populates a cache. Subsequent requests with the same hash value are sent to the same service. The following flow chart illustrates this process.

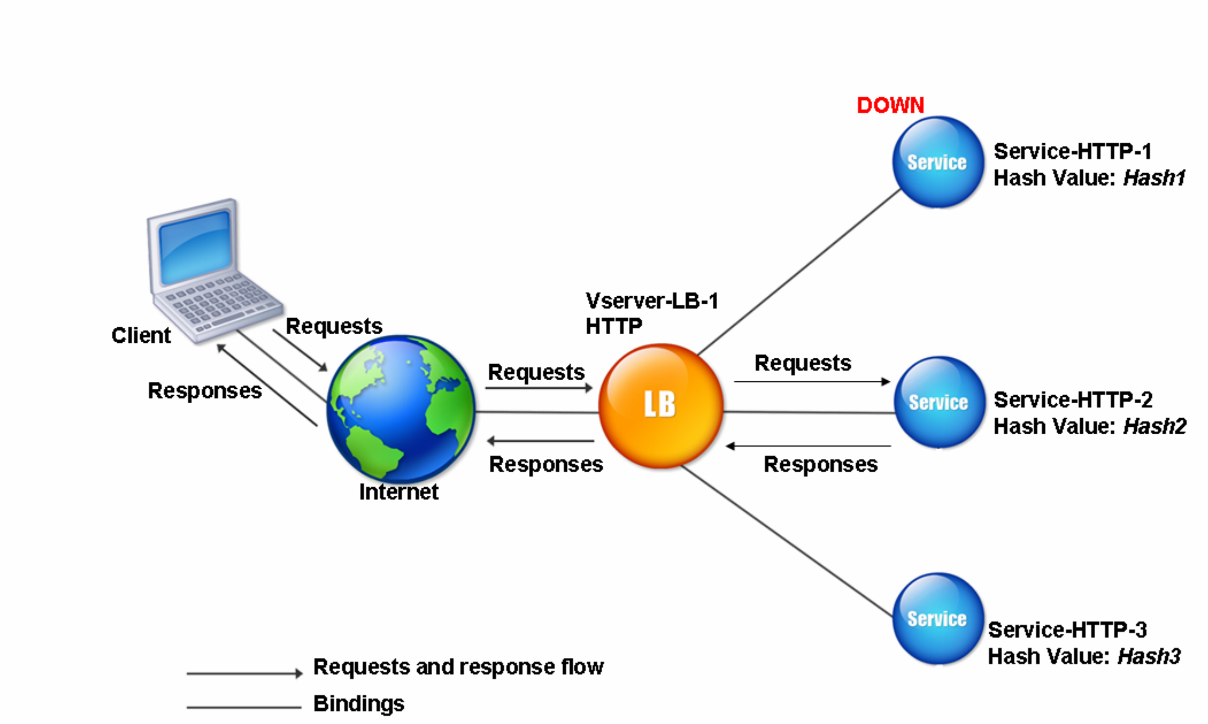
Figure 1. How the Hashing Methods Distribute Requests


NetScaler computes Hash value by using the configured Hash Methods
Compute Hash Value by using all services
Select a Service with highest hash value from the service list
Is the state of the Service UP?
NO
YES
Forward the request to the selected service

Hashing methods can be applied to IPv4 and IPv6 addresses.

Consider a scenario where three services (Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3) are bound to a virtual server, any hash method is configured, and the hash value is Hash1. When the configured services are UP, the request is sent to Service-HTTP-1. If Service-HTTP-1 is down, the NetScaler appliance calculates the hash value for the last log of the number of services. The appliance then selects the service with the highest hash value, such as Service-HTTP-2. The following diagram illustrates this process.

Figure 2. Entity Model for Hashing Methods


DOWN
Service
Service-HTTP-1
Hash Value: *Hash1*
Client
Requests
Responses
Internet
Vserver-LB-1
HTTP
Requests
Responses
LB
Requests
Responses
Service
Service-HTTP-2
Hash Value: *Hash2*
Service
Service-HTTP-3
Hash Value: *Hash3*
Requests and response flow
Bindings

Note: If the NetScaler appliance fails to select a service by using a hashing method, it defaults to the least connection method to select a service for the incoming request. You should adjust server pools by removing services during periods of low traffic to enable the caches to repopulate without affecting performance on your load balancing setup.

The URL Hash Method

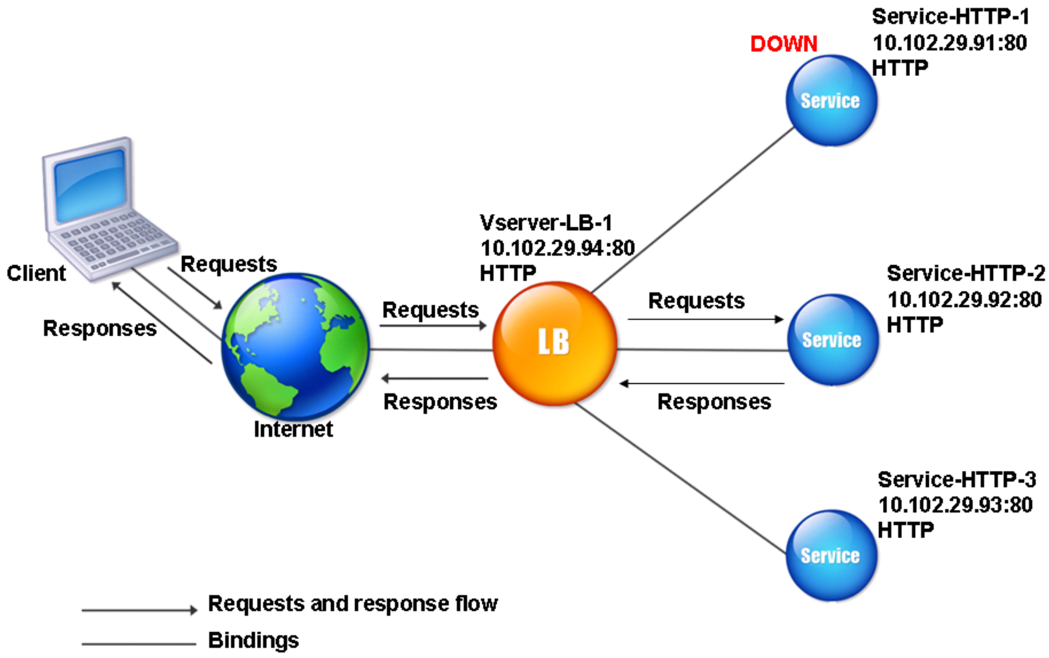
When you configure the NetScaler appliance to use the URL hash method for load balancing the services, for selecting a service, the appliance generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The appliance caches the hashed value of the URL, and when it receives subsequent requests that use the same URL, it forwards them to the same service. If the appliance cannot parse an incoming request, it uses the round robin method for load balancing instead of the URL hash method.

For generating the hash value, the appliance uses a specific algorithm and considers a part of the URL. By default, the appliance considers the first 80 bytes of the URL. If the URL is of less than 80 bytes, the complete URL is used. You can specify a different length. The hash length can be from 1 to 4096 bytes. Generally, if long URLs are used where only a small number of characters are different, it is a good idea to make the hash length as high as possible to try to ensure a more even load distribution.

Consider a scenario where three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3, are bound to a virtual server, and the load balancing method configured on the virtual server is the URL hash method. The virtual server receives a request and the hash value of the URL is U1. appliance selects Service-HTTP-1. If Service-HTTP-1 is DOWN, the appliance selects Service-HTTP-2.

The following diagram illustrates this process.

Figure 3. How URL Hashing Operates


DOWN
Service-HTTP-1
10.102.29.91:80
HTTP
Service
Client
Requests
Responses
Internet
Vserver-LB-1
10.102.29.94:80
HTTP
Requests
Responses
LB
Requests
Responses
Service-HTTP-2
10.102.29.92:80
HTTP
Service
Service-HTTP-3
10.102.29.93:80
HTTP
Service
Requests and response flow
Bindings

If both Service-HTTP-1 and Service-HTTP-2 are DOWN, appliance sends requests with hash value U1 to Service-HTTP-3.

If Service-HTTP-1 and Service-HTTP-2 are down, requests that generate the hash URL1 are sent to Service-HTTP-3. If these services are UP, requests that generate the hash URL1 are distributed in the following manner:

- If the Service-HTTP-2 is up, the request is sent to Service-HTTP-2.
- If the Service-HTTP-1 is up, the request is sent to Service-HTTP-1.
- If Service-HTTP-1 and Service-HTTP-2 are up at the same time, the request is sent to Service-HTTP-1.

To configure the URL hash method, see Configuring a Load Balancing Method that Does Not Include a Policy. Select the load balancing method as URL Hash, and set the hash length to the number of bytes to be used for generating the hash value.

The Domain Hash Method

A load balancing virtual server configured to use the domain hash method uses the hashed value of the domain name in the HTTP request to select a service. The domain name is taken from either the incoming URL or the Host header of the HTTP request. If the domain name appears in both the URL and the Host header, the appliance gives preference to the URL.

If you configure domain name hashing, and an incoming HTTP request does not contain a domain name, the NetScaler appliance defaults to the round robin method for that request.

The hash-value calculation uses the name length or hash length value, whichever is smaller. By default, the NetScaler appliance calculates the hash value from the first 80 bytes of the domain name. To specify a different number of bytes in the domain name when calculating the hash value, you can set the hashLength parameter (Hash Length in the configuration utility) to a value of from 1 to 4096 (bytes).

To configure the domain hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Destination IP Hash Method

A load balancing virtual server configured to use the destination IP hash method uses the hashed value of the destination IP address to select a server. You can mask the destination IP address to specify which part of it to use in the hash value calculation, so that requests that are from different networks but destined for the same subnet are all directed to the same server. This method supports IPv4 and IPv6-based destination servers.

This load balancing method is appropriate for use with the cache redirection feature.

To configure the destination IP hash method for an IPv4 destination server, you set the netMask parameter. To configure this method for an IPv6 destination server, you use the v6NetMaskLen parameter. In the configuration utility, text boxes for setting these parameters appear when you select the Destination IP Hash method.

To configure the destination IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Hash Method

A load balancing virtual server configured to use the source IP hash method uses the hashed value of the client IPv4 or IPv6 address to select a service. To direct all requests from source IP addresses that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Destination IP Hash Method

A load balancing virtual server configured to use the source IP destination IP hash method uses the hashed value of the source and destination IP addresses (either IPv4 or IPv6) to select a service. Hashing is symmetric; the hash-value is the same regardless of the order of the source and destination IPs. This ensures that all packets flowing from a particular client to the same destination are directed to the same server.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP destination IP hash method, see see Configuring a Load Balancing Method that Does Not Include a Policy.

The Source IP Source Port Hash Method

A load balancing virtual server configured to use the source IP source port hash method uses the hash value of the source IP (either IPv4 or IPv6) and source port to select a service. This ensures that all packets on a particular connection are directed to the same service.

This method is used in connection mirroring and firewall load balancing. For more information about connection mirroring, see Connection Failover.

To direct all requests that belong to a particular network to a specific destination server, you must mask the source IP address. For IPv4 addresses, use the netMask parameter. For IPv6 addresses, use the v6NetMaskLength parameter.

To configure the source IP source port hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

The Call ID Hash Method

A load balancing virtual server configured to use the call ID hash method uses the hash value of the call ID in the SIP header to select a service. Packets for a particular SIP session are therefore always directed to the same proxy server.

This method is applicable to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure the call ID hash method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least bandwidth method

September 21, 2020

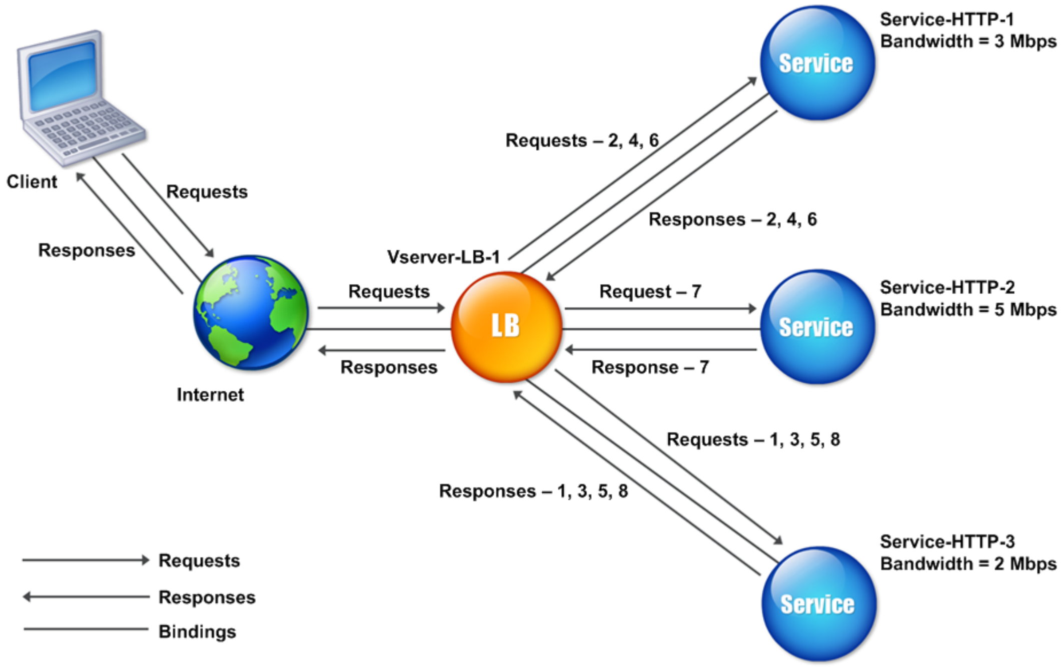
A load balancing virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps). The following example shows how the virtual server selects a service for load balancing by using the least bandwidth method.

Consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has 3 Mbps bandwidth.
- Service-HTTP-2 has 5 Mbps bandwidth.
- Service-HTTP-3 has 2 Mbps bandwidth.

The following diagram illustrates how the virtual server uses the least bandwidth method to forward requests to the three services.

Figure 1. How the Least Bandwidth Load Balancing Method Works


Client
Requests
Responses
Internet
Requests
Responses
Vserver-LB-1
LB
Requests – 2, 4, 6
Responses – 2, 4, 6
Request – 7
Response – 7
Requests – 1, 3, 5, 8
Responses – 1, 3, 5, 8
Service
Service-HTTP-1
Bandwidth = 3 Mbps
Service
Service-HTTP-2
Bandwidth = 5 Mbps
Service
Service-HTTP-3
Bandwidth = 2 Mbps
Requests
Responses
Bindings

The virtual server selects the service by using the bandwidth value (N), which is the sum of the number of bytes transmitted and received over the previous 14 seconds. If each request requires 1 Mbps bandwidth, the NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have same N value, the virtual server switches to the round robin method for these servers, alternating between them. Service-HTTP-1 receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 now all have same N value, the virtual server includes Service-HTTP-2 in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.

Request Received	Service Selected	Current N Value	Remarks
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3;(N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler appliance uses the number of data and control bytes exchanged to determine the bandwidth usage for RTSP services. For more information about RTSP NAT option, see
Managing RTSP Connections.

The NetScaler appliance also performs load balancing by using the bandwidth and weights if different weights are assigned to the services. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.

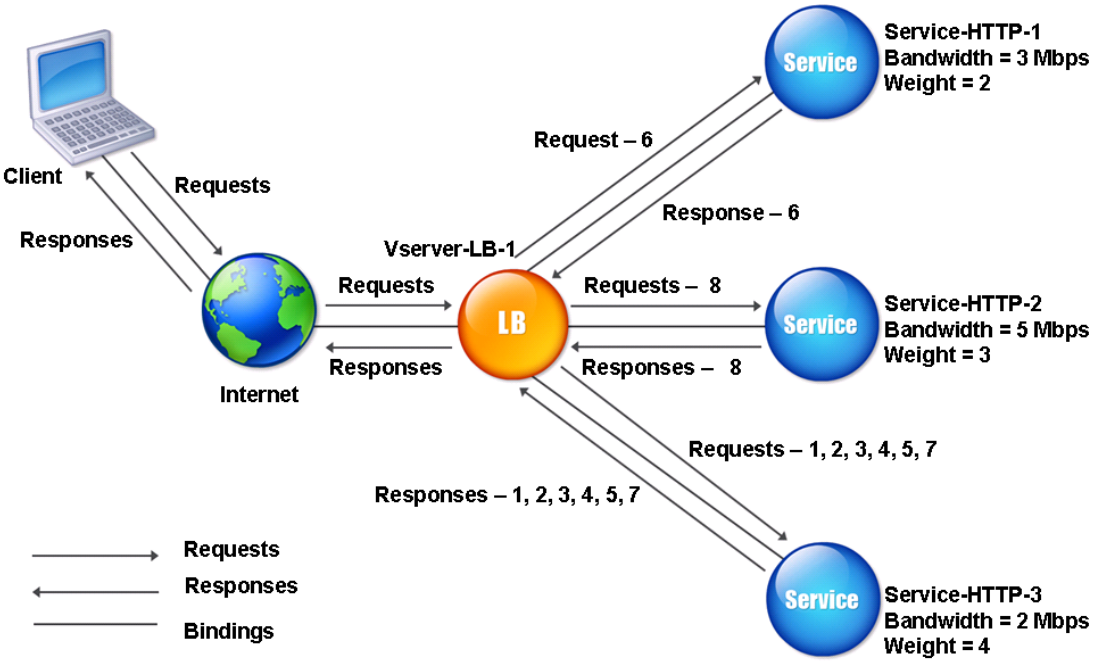
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3; (Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least bandwidth method when weights are assigned to the services.

Figure 2. How the Least Bandwidth Load Balancing Method Works When Weights Are Assigned


Client
Requests
Responses
Internet
Vserver-LB-1
Requests
Responses
LB
Request – 6
Response – 6
Requests – 8
Responses – 8
Requests – 1, 2, 3, 4, 5, 7
Responses – 1, 2, 3, 4, 5, 7
Service
Service-HTTP-1
Bandwidth = 3 Mbps
Weight = 2
Service
Service-HTTP-2
Bandwidth = 5 Mbps
Weight = 3
Service
Service-HTTP-3
Bandwidth = 2 Mbps
Weight = 4
Requests
Responses
Bindings

To configure the least bandwidth method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Least packets method

September 21, 2020

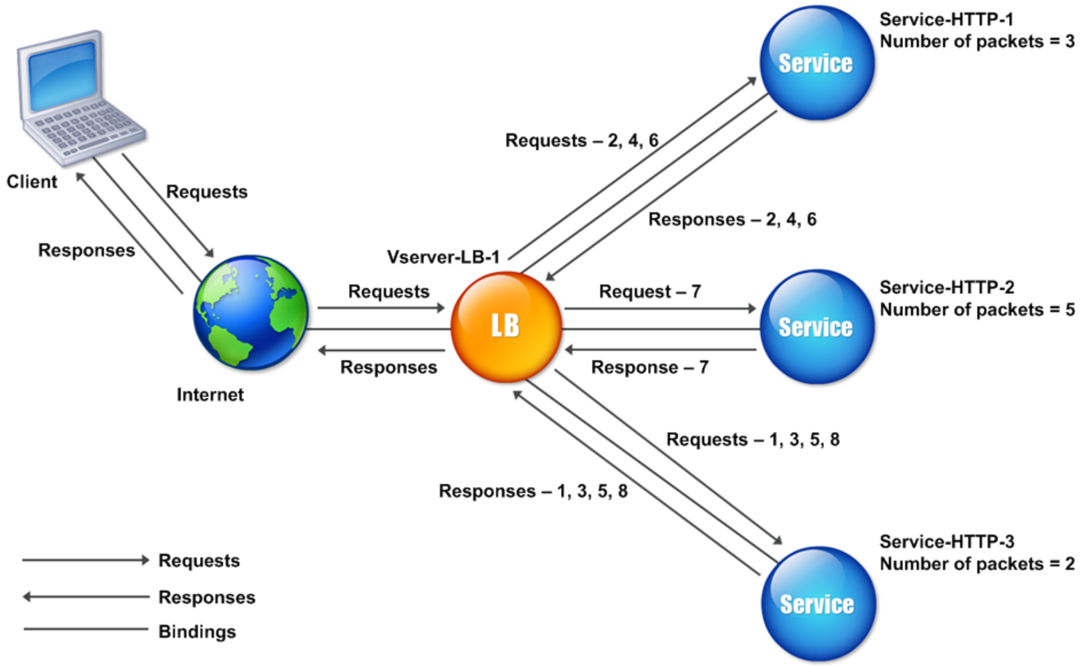
A load balancing virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 has handled three packets in last 14 seconds.
- Service-HTTP-2 has handled five packets in last 14 seconds.
- Service-HTTP-3 has handled two packets in last 14 seconds.

The following diagram illustrates how the NetScaler appliance uses the least packets method to choose a service for each request that it receives.

Figure 1. How the Least Packets Load Balancing Method Works


Client
Requests
Responses
Internet
Requests
Responses
Vserver-LB-1
LB
Requests – 2, 4, 6
Responses – 2, 4, 6
Request – 7
Response – 7
Requests – 1, 3, 5, 8
Responses – 1, 3, 5, 8
Service
Service-HTTP-1
Number of packets = 3
Service
Service-HTTP-2
Number of packets = 5
Service
Service-HTTP-3
Number of packets = 2
Requests
Responses
Bindings

The NetScaler appliance selects a service by using the number of packets (N) transmitted and received by each service in the last 14 seconds. Using this method, it delivers requests as follows:

- Service-HTTP-3 receives the first request, because this service has the lowest N value.
- Since Service-HTTP-1 and Service-HTTP-3 now have the same N value, the virtual server switches to the round robin method. Service-HTTP-1 therefore receives the second request, Service-HTTP-3 receives the third request, Service-HTTP-1 receives the fourth request, Service-HTTP-3 receives the fifth request, and Service-HTTP-1 receives the sixth request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same N value, the virtual server switches to the round robin method for Service-HTTP-2 as well, including it in the round robin list. Therefore, Service-HTTP-2 receives the seventh request, Service-HTTP-3 receives the eighth request, and so on.

The following table summarizes how N is calculated.

Request Received	Service Selected	Current N Value	Remarks
Request-1	Service-HTTP-3; (N = 2)	N = 3	Service-HTTP-3 has the lowest N value.

Request Received	Service Selected	Current N Value	Remarks
Request-2	Service-HTTP-1; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-3	Service-HTTP-3; (N = 3)	N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-4	Service-HTTP-1; (N = 4)	N = 5	-
Request-5	Service-HTTP-3; (N = 4)	N = 5	-
Request-6	Service-HTTP-1; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 5)	N = 6	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-3; (N = 5)	N = 6	-

Note: If you enable the RTSP NAT option on the virtual server, the applianace uses the number of data and control packets to calculate the number of packets for RTSP services. For more information about RTSP NAT option, see
Managing RTSP Connections.

The NetScaler appliance also performs load balancing by using the number of packets and weights when a different weight is assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 2, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 4. The NetScaler appliance delivers requests as follows:

- Service-HTTP-3 receives the first second, third, fourth, and fifth requests, because this service has the lowest Nw value.

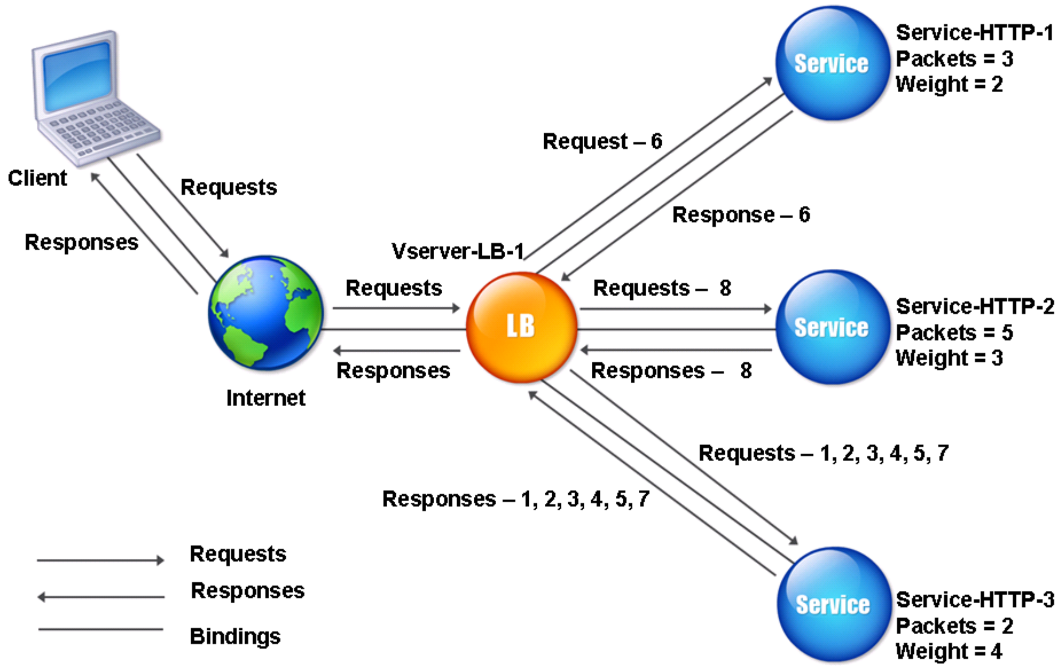
- Service-HTTP-1 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the seventh request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the eighth request, because this service has the lowest Nw value.

The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-1	Service-HTTP-3; (Nw = 5000)	Nw = 5000	Service-HTTP-3 has the lowest Nw value.
Request-2	Service-HTTP-3; (Nw = 5000)	Nw = 7500	-
Request-3	Service-HTTP-3;(Nw = 7500)	Nw = 10000	-
Request-4	Service-HTTP-3; (Nw = 10000)	Nw = 12500	-
Request-5	Service-HTTP-3; (Nw = 12500)	Nw = 15000	-
Request-6	Service-HTTP-1; (Nw = 15000)	Nw = 20000	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-7	Service-HTTP-3; (Nw = 15000)	Nw = 17500	Service-HTTP-1 and Service-HTTP-3 have the same Nw value.
Request-8	Service-HTTP-2; (Nw = 16666.67)	Nw = 20000	Service-HTTP-2 has the lowest Nw value.

The following diagram illustrates how the virtual server uses the least packets method when weights are assigned.

Figure 2. How the Least Packets Method Works When Weights Are Assigned


Client
Requests
Responses
Internet
Requests
Responses
Vserver-LB-1
LB
Request – 6
Response – 6
Service
Service-HTTP-1
Packets = 3
Weight = 2
Requests – 8
Responses – 8
Service
Service-HTTP-2
Packets = 5
Weight = 3
Requests – 1, 2, 3, 4, 5, 7
Responses – 1, 2, 3, 4, 5, 7
Service
Service-HTTP-3
Packets = 2
Weight = 4
Requests
Responses
Bindings

To configure the least packets method, see Configuring a Load Balancing Method that Does Not Include a Policy.

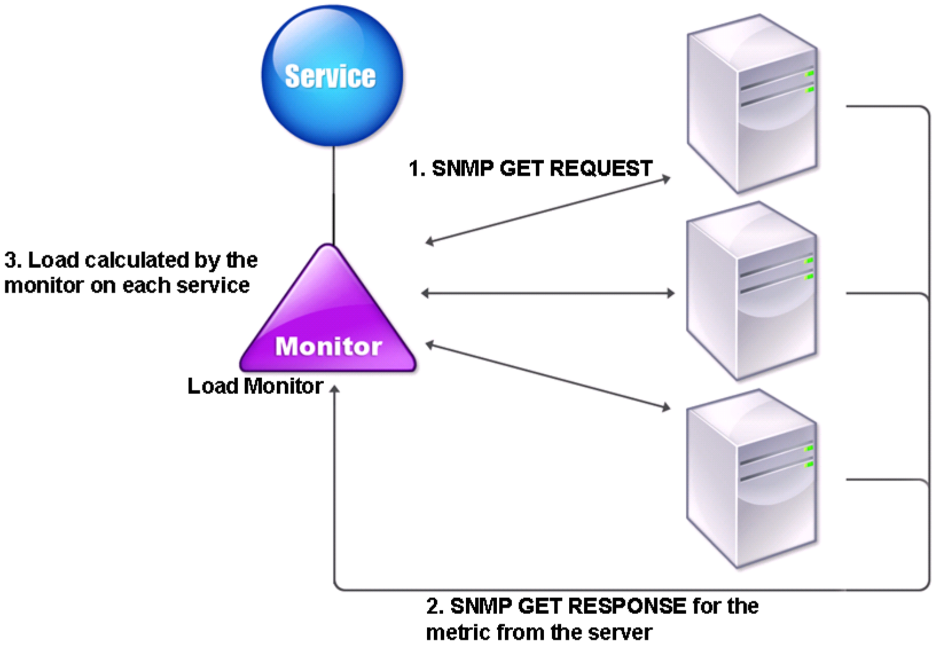
Custom load method

September 21, 2020

Custom load balancing is performed on server parameters such as CPU usage, memory, and response time. When using the custom load method, the NetScaler appliance usually selects a service that is not handling any active transactions. If all of the services in the load balancing setup are handling active transactions, the appliance selects the service with the smallest load. A special type of monitor, known as a load monitor, calculates the load on each service in the network. The load monitors do not mark the state of a service, but they do take services out of the load balancing decision when those services are not UP.

For more information about load monitors, see Understanding Load Monitors. The following diagram illustrates how a load monitor operates.

Figure 1. How Load Monitors Operate


Service
1. SNMP GET REQUEST
3. Load calculated by the monitor on each service
Monitor
Load Monitor
2. SNMP GET RESPONSE for the metric from the server

The load monitor uses Simple Network Management Protocol (SNMP) probes to calculate load on each service by sending an SNMP GET request to the service. This request contains one or more object IDs (OIDs). The service responds with an SNMP GET response, with metrics corresponding to the SNMP OIDs. The load monitor uses the response metrics, described below, to calculate the load on the service.

The load monitor calculates the load on a service by using the following parameters:

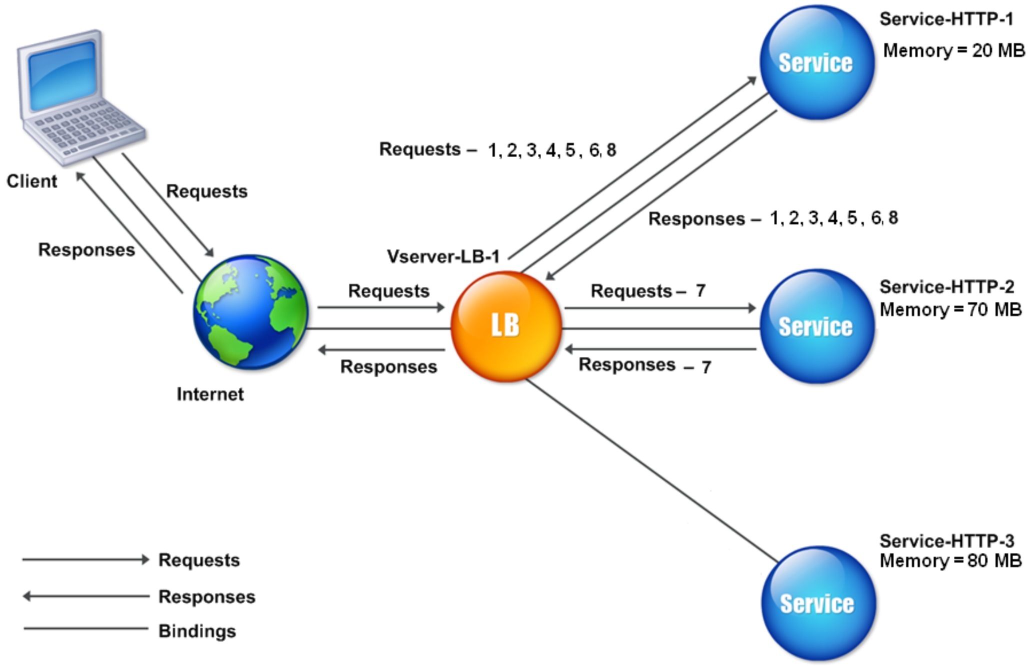
- Metrics values retrieved through SNMP probes that exist as tables in the NetScaler appliance.
- Threshold value set for each metric.
- Weight assigned to each metric.

For example, consider three services, Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3.

- Service-HTTP-1 is using 20 megabytes (MB) of memory.
- Service-HTTP-2 is using 70 MB of memory.
- Service-HTTP-3 is using 80 MB of memory.

The load balanced servers can export metrics such as CPU and memory usage to the services, which can in turn provide them to the load monitor. The load monitor sends an SNMP GET request containing the OIDs 1.3.6.1.4.1.5951.4.1.1.41.1.5, 1.3.6.1.4.1.5951.4.1.1.41.1.4, and 1.3.6.1.4.1.5951.4.1.1.41.1.3 to the services. SNMP OIDs of type STRING are not supported, because you cannot calculate the load by using a STRING OID. Loads can be calculated by using other data types, such as INT and gauge32. The three services respond to the request. The NetScaler appliance compares the exported metrics, and then selects Service-HTTP-1 because it has more available memory. The following diagram illustrates this process.

Figure 2. How the Custom Load Method Works


Service-HTTP-1
Memory = 20 MB
Service
Requests – 1, 2, 3, 4, 5, 6, 8
Client
Requests
Responses – 1, 2, 3, 4, 5, 6, 8
Responses
Vserver-LB-1
Requests
LB
Requests – 7
Service-HTTP-2
Memory = 70 MB
Service
Responses
Responses – 7
Internet
Service-HTTP-3
Memory = 80 MB
Service
Requests
Responses
Bindings

If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, and fifth requests, because this service has the lowest N value.
- Service-HTTP-1 and Service-HTTP-2 now have the same load, so the virtual server reverts to the round robin method for these servers. Therefore, Service-HTTP-2 receives the sixth request, and Service-HTTP-1 receives the seventh request.
- Since Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 all now have same load, the virtual server reverts to the round robin method for Service-HTTP-3 as well. Therefore, Service-HTTP-3 receives the eighth request.

The following table summarizes how N is calculated.

Request received	Service selected	Current N Value (Number of Active Transactions)	Remarks
Request-1	Service-HTTP-1; (N = 20)	N = 30	Service-HTTP-3 has the lowest N value.
Request-2	Service-HTTP-1; (N = 30)	N = 40	-
Request-3	Service-HTTP-1; (N = 40)	N = 50	-
Request-4	Service-HTTP-1; (N = 50)	N = 60	-
Request-5	Service-HTTP-1; (N = 60)	N = 70	-
Request-6	Service-HTTP-1; (N = 70)	N = 80	Service-HTTP-2 and Service-HTTP-3 have the same N values.
Request-7	Service-HTTP-2; (N = 70)	N = 80	Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-1; (N = 80)	N = 90	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.

If different weights are assigned to the services, the custom load algorithm considers both the load on each service and the weight assigned to each service. It selects a service by using the value (Nw) in the following expression:

Nw = (N) * (10000 / weight)

As in the preceding example, suppose Service-HTTP-1 is assigned a weight of 4, Service-HTTP-2 is assigned a weight of 3, and Service-HTTP-3 is assigned a weight of 2. If each request uses 10 MB memory, the NetScaler appliance delivers requests as follows:

- Service-HTTP-1 receives the first, second, third, fourth, fifth, sixth, seventh, and eighth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the ninth request, because this service has the lowest Nw value.

Service-HTTP-3 has the highest Nw value, and is therefore not considered for load balancing.

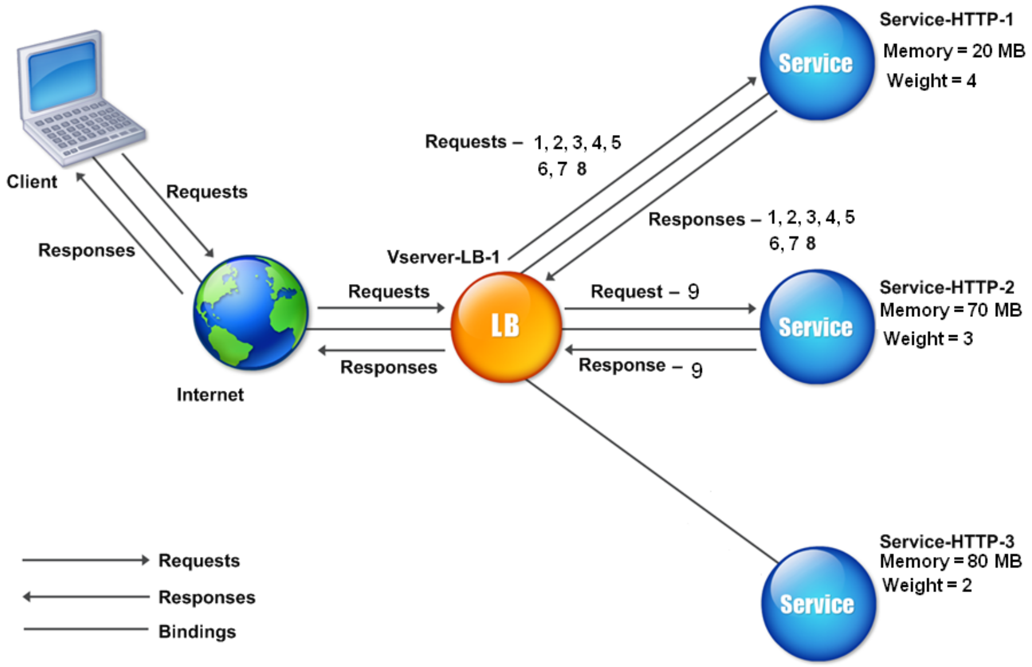
The following table summarizes how Nw is calculated.

Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-1; (Nw = 50000)	Nw = 75000	Service-HTTP-1 has the lowest Nw value.
Request-2	Service-HTTP-1; (Nw = 5000)	Nw = 100000	-
Request-3	Service-HTTP-1; (Nw = 15000)	Nw = 125000	-
Request-4	Service-HTTP-1; (Nw = 20000)	Nw = 150000	-
Request-5	Service-HTTP-1; (Nw = 23333.34)	Nw = 175000	-
Request-6	Service-HTTP-1; (Nw = 25000)	Nw = 200000	-
Request-7	Service-HTTP-1; (Nw = 23333.34)	Nw = 225000	-
Request-8	Service-HTTP-1; (Nw = 25000)	Nw = 250000	-
Request-9	Service-HTTP-2; (Nw = 233333.34)	Nw = 266666.67	Service-HTTP-2 has the lowest Nw value.

Service-HTTP-1 is selected for load balancing when it completes its active transactions or when the Nw value of other services (Service-HTTP-2 and Service-HTTP-3) is equal to 400,000.

The following diagram illustrates how the NetScaler appliance uses the custom load method when weights are assigned.

Figure 3. How the Custom Load Method Works When Weights Are Assigned


Client
Requests
Responses
Internet
Requests
Responses
Vserver-LB-1
LB
Requests – 1, 2, 3, 4, 5
6, 7 8
Responses – 1, 2, 3, 4, 5
6, 7 8
Request – 9
Response – 9
Service
Service-HTTP-1
Memory = 20 MB
Weight = 4
Service
Service-HTTP-2
Memory = 70 MB
Weight = 3
Service
Service-HTTP-3
Memory = 80 MB
Weight = 2
Requests
Responses
Bindings

To configure the custom load method, see Configuring a Load Balancing Method that Does Not Include a Policy.

Static proximity method

September 21, 2020

When a virtual server is configured to use the static proximity method, it selects the service that best matches the proximity criteria.

For the static proximity method to work, you must either configure the NetScaler appliance to use an existing static proximity database populated through a location file or add custom entries to the static proximity database. After adding custom entries, you can set their location qualifiers. After configuring the database, you are ready to specify static proximity as the load balancing method.

For more details, see the following topics.

- Adding a Location File to Create a Static Proximity Database
- Adding Custom Entries to a Static Proximity Database
- Setting the Location Qualifiers

- Specifying the Static Proximity method

Specifying the Proximity Method

When you have configured the static proximity database, you are ready to specify static proximity as the GLSB method.

To specify static proximity by using the command line interface

At the command prompt, type the following commands to configure static proximity and verify the configuration:

```
set lb vserver <name> -lbMethod STATICPROXIMITY

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod STATICPROXIMITY

show lb vserver
<!--NeedCopy-->
```

To specify static proximity by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers** and select the virtual server.
2. Click **Edit** and expand the **Method** section.
3. In the **Load Balancing Method** list, select **STATICPROXIMITY**.

Token method

September 21, 2020

A load balancing virtual server configured to use the token method bases its selection of a service on the value of a data segment extracted from the client request. The data segment is called the token. You configure the location and size of the token. For subsequent requests with the same token, the virtual server chooses the same service that handled the initial request.

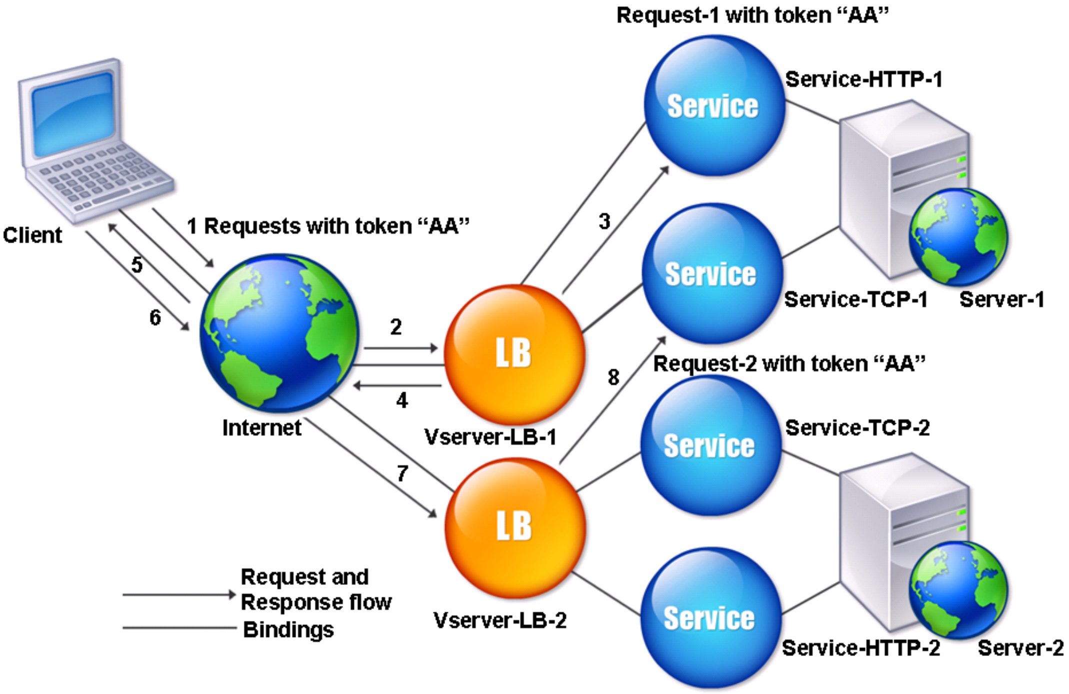
This method is content aware; it operates differently for TCP, HTTP, and HTTPS connections. For HTTP or HTTPS services, the token is found in the HTTP headers, the URL, or the BODY. To locate the token, you specify or create a classic or advanced expression. For more information on classic or advanced expressions, see Policy Configuration and Reference.

For HTTP services, the virtual server searches for the configured token in the first 24 kilobytes (KB) of the TCP payload. For non-HTTP (TCP, SSL, and SSL_TCP) services, the virtual server searches for the configured token in the first 16 packets if the total size of the 16 packets is less than 24 KB. But if the total size of the 16 packets is greater than 24 KB, the appliance searches for the token in the first 24 KB of payload. You can use this load balancing method across virtual servers of different types to make sure that requests presenting the same token are directed to appropriate services, regardless of the protocol used.

For example, consider a load balancing setup consisting of servers that contain Web content. You want to configure the NetScaler appliance to search for a specific string (the token) inside the URL query portion of the request. Server-1 has two services, Service-HTTP-1 and Service-TCP-1, and Server-2 has two services, Service-HTTP-2 and Service-TCP-2. The TCP services are bound to Vserver-LB-2, and the HTTP services are bound to Vserver-LB-1.

If Vserver-LB-1 receives a request with the token AA, it selects the service Service-HTTP-1 (bound to server-1) to process the request. If Vserver-LB-2 receives a different request with the same token (AA), it directs this request to the service Service-TCP-1. The following diagram illustrates this process.

Figure 1. How the Token Method Works


Request-1 with token "AA"
Service-HTTP-1
Service
Client
1 Requests with token "AA"
3
5
6
2
4
Service
Service-TCP-1
Server-1
LB
8
Request-2 with token "AA"
Internet
Vserver-LB-1
Service
Service-TCP-2
7
LB
Request and Response flow
Bindings
Vserver-LB-2
Service
Service-HTTP-2
Server-2

To configure the Token load balancing method by using the command line interface

At the command prompt, type the following commands to configure the token load balancing method and verify the configuration:

```
set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length>
    -dataoffset <offset>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -
   dataoffset 25

show lb vserver LB-VServer-1
<!--NeedCopy-->
```

To configure the token load balancing method by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, click Method
3. In the Load Balancing Method list, select Token, and specify an expression.

Configure a load balancing method that does not include a policy

September 21, 2020

After you select a load balancing algorithm for your load balancing setup, you must configure the NetScaler appliance to use that algorithm. You can configure it by using the CLI or by using the configuration utility.

Note:

The token method is policy based and requires more configuration than is described here. To configure the token method, see Token method.

For some hash-based methods, you can mask an IP address to direct requests belonging to the same subnet to the same server. For more information, see Hashing methods.

To set the load balancing method by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -lbMethod <method>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
<!--NeedCopy-->
```

To set the load balancing method by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, click **Method**, and in the Load Balancing Method list, select a method.

Persistence and persistent connections

September 21, 2020

Unless you configure persistence, a load balancing stateless protocol, such as HTTP, disrupts the maintenance of state information about client connections. Different transmissions from the same client might be directed to different servers even though all of the transmissions are part of the same session. You must configure persistence on a load balancing virtual server that handles certain types of Web applications, such as shopping cart applications.

Before you can configure persistence, you need to understand the different types of persistence, how they are used, and what the implications of each type is. You then need to configure the NetScaler appliance to provide persistent connections for those Web sites and Web applications that require them.

You can also configure backup persistence, which takes effect in the event that the primary type of persistence configured for a load balancing virtual server fails. You can configure persistence groups, so that a client transmission to any virtual server in a group can be directed to a server that has received previous transmissions from the same client.

For information about persistence with RADIUS load balancing, see Configuring RADIUS Load Balancing with Persistence.

About Persistence

September 21, 2020

You can choose from among any of several types of persistence for a given load balancing virtual server, which then routes to the same service all connections from the same user to your shopping cart application, Web-based email, or other network application. The persistence session remains in effect for a period of time, which you specify.

If a server participating in a persistence session goes DOWN, the load balancing virtual server uses the configured load balancing method to select a new service, and establishes a new persistence session with the server represented by that service. If the server goes OUT OF SERVICE, it continues to process existing persistence sessions, but the virtual server does not direct any new traffic to it. After the shutdown period elapses, the virtual server ceases to direct connections from existing clients to the service, closes existing connections, and redirects those clients to new services if necessary.

Depending on the persistence type you configure, the NetScaler appliance might examine the source IPs, destination IPs, SSL session IDs, Host or URL headers, or some combination of these things to place each connection in the proper persistence session. It might also base persistence on a cookie issued by the Web server, on an arbitrarily assigned token, or on a logical rule. Almost anything that allows the appliance to match connections with the proper persistence session and be used as the basis for persistence.

The following table summarizes the persistence types available on the NetScaler appliance.

Persistence Type	Description
Source IP	SOURCEIP. Connections from the same client IP address are parts of the same persistence session.
HTTP Cookie	COOKIEINSERT. Connections that have the same HTTP Cookie header are parts of the same persistence session.
SSL Session ID	SSLSESSION. Connections that have the same SSL Session ID are parts of the same persistence session.
URL Passive	URLPASSIVE. Connections to the same URL are treated as parts of the same persistence session.
Custom Server ID	CUSTOMSERVERID. Connections with the same HTTP HOST header are treated as parts of the same persistence session.
Destination IP	DESTIP. Connections to the same destination IP are treated as parts of the same persistence session.
Source and Destination IPs	SRCIPDESTIP. Connections that are both from the same source IP and to the same destination IP are treated as parts of the same persistence session.
SIP Call ID	CALLID. Connections that have the same call ID in the SIP header are treated as parts of the same persistence session.
RTSP Session ID	RTSPSID. Connections that have the same RTSP Session ID are treated as parts of the same persistence session.

Persistence Type	Description
User-Defined Rule	RULE. Connections that match a user-defined rule are treated as parts of the same persistence session.

Table 1. Types of Persistence

Depending on the type of persistence that you have configured, the virtual server can support either 250,000 simultaneous persistent connections or any number of persistent connections up to the limits imposed by the amount of RAM on your NetScaler appliance. The following table shows which types of persistence fall into each category.

Persistence Type	Number of Simultaneous Persistent Connections Supported
Source IP, SSL Session ID, Rule, destination IP, source IP/destination IP, SIP Call ID, RTSP Session ID	250 K
Cookie, URL Server ID, Custom Server ID	Memory limit. In case of CookieInsert, if timeout is not 0, the number of connections is limited by memory.

Table 2. Persistence Types and Numbers of Simultaneous Connections Supported

Some types of persistence are specific to particular types of virtual server. The following table lists each type of persistence and indicates which types of persistence are supported on which types of virtual server.

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
SOURCEI	YES	YES	YES	YES	YES	YES	NO	NO
COOKIEINSERT	YES	YES	NO	NO	NO	NO	NO	NO
SSLSESS	NO	YES	NO	NO	YES	YES	NO	NO
URLPASSIVE	YES	YES	NO	NO	NO	NO	NO	NO
CUSTOM	YES	YES	NO	NO	NO	NO	NO	NO
RULE	YES	YES	YES	NO	NO		NO	NO
SRCIPDE	YES	YES	YES	YES	YES	YES	NO	NO

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP	SIP_UDP
DESTIP	YES	YES	YES	YES	YES	YES	NO	NO
CALLID	NO	NO	NO	NO	NO	NO	NO	YES
RTSPID	NO	NO	NO	NO	NO	NO	YES	NO

Table 3. Relationship of Persistence Type to Virtual Server Type

Source IP address persistence

September 21, 2020

When source IP persistence is configured, the load balancing virtual server uses the configured load balancing method to select a service for the initial request, and then uses the source IP address (client IP address) to identify subsequent requests from that client and send them to the same service. You can set a time-out value, which specifies the maximum inactivity period for the session. When the time-out value expires, the session is discarded, and the configured load balancing algorithm is used to select a new server.

Caution: In some circumstances, using persistence based on source IP address can overload your servers. All requests to a single Web site or application are routed through the single gateway to the NetScaler appliance, even though they are then redirected to multiple locations. In multiple proxy environments, client requests frequently have different source IP addresses even when they are sent from the same client, resulting in rapid multiplication of persistence sessions where a single session should be created. This issue is called the “Mega Proxy problem.” You can use HTTP cookie-based persistence instead of Source IP-based persistence to prevent this from happening.

To configure persistence based on Source IP Address, see Configuring Persistence Types That Do Not Require a Rule.

Note: If all incoming traffic comes from behind a Network Address Translation (NAT) device or proxy, the traffic appears to the NetScaler appliance to come from a single source IP address. This prevents Source IP persistence from functioning properly. Where this is the case, you must select a different persistence type.

HTTP cookie persistence

September 21, 2020

When HTTP cookie persistence is configured, the NetScaler appliance sets a cookie in the HTTP headers of the initial client request. The cookie contains the IP address and port of the service selected by the load balancing algorithm. As with any HTTP connection, the client then includes that cookie with any subsequent requests.

When the NetScaler appliance detects the cookie, it forwards the request to the service IP and port in the cookie, maintaining persistence for the connection. You can use this type of persistence with virtual servers of type HTTP or HTTPS. This persistence type does not consume any appliance resources and therefore can accommodate an unlimited number of persistent clients.

Note: If the client's Web browser is configured to refuse cookies, HTTP cookie-based persistence will not work. It might be advisable to configure a cookie check on the Web site, and warn clients that do not appear to be storing cookies properly that they will need to enable cookies for the Web site if they want to use it.

The format of the cookie that the NetScaler appliance inserts is:

```
NSC_XXXX=<ServiceIP ><ServicePort>
```

where:

- NSC_XXXX is the virtual server ID that is derived from the virtual server name.
- ServiceIP and ServicePort are encoded representations of the service IP address and service port, respectively. The IP address and port are encoded separately.

You can set a time-out value for this type of persistence to specify an inactivity period for the session. When the connection has been inactive for the specified period of time, the NetScaler appliance discards the persistence session. Any subsequent connection from the same client results in a new server being selected based on the configured load balancing method, and a new persistence session being established.

Note: If you set the time-out value to 0, the NetScaler appliance does not specify an expiration time, but sets a session cookie that is not saved when the client's browser is shut down.

By default, the NetScaler appliance sets HTTP version 0 cookies for maximum compatibility with client browsers. (Only certain HTTP proxies understand version 1 cookies; most commonly used browsers do not.) You can configure the appliance to set HTTP version 1 cookies, for compliance with RFC2109. For HTTP version 0 cookies, the appliance inserts the cookie expiration date and time as an absolute Coordinated Universal Time (GMT). It calculates this value as the sum of the current GMT time on the appliance and the time-out value. For HTTP version 1 cookies, the appliance inserts a relative expiration time by setting the "Max-Age" attribute of the HTTP cookie. In this case, the client's browser calculates the actual expiration time.

To configure persistence based on a cookie inserted by the appliance, see Configuring Persistence Types That Do Not Require a Rule.

In the HTTP cookie, the appliance by default sets the httponly flag to indicate that the cookie is non-scriptable and should not be revealed to the client application. Therefore, a client-side script cannot access the cookie, and the client is not susceptible to cross-site scripting.

Certain browsers, however, do not support the httponly flag and, therefore, might not return the cookie. As a result, persistence is broken. For browsers that do not support the flag, you can omit the httponly flag in the persistence cookie.

To change the httponly flag setting by using the command line interface

At the command prompt, type:

```
set lb parameter -httpOnlyCookieFlag (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
>  set lb parameter -httpOnlyCookieFlag disabled
   Done
> show lb parameter
  Global LB parameters:
       Persistence Cookie HttpOnly Flag: DISABLED
       Use port for hash LB: YES
 Done
<!--NeedCopy-->
```

To change the httponly flag setting by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters, and select or clear the Persistence Cookie HTTPOnly Flag.

Encrypting the Cookie

From release 10.5 build 55.8, you can encrypt the cookie in addition to any SSL encryption.

To encrypt the cookie by using the command line interface, at the command prompt, type

```
set lb parameter -UseEncryptedPersistenceCookie ENABLED -
    cookiePassphrase test
<!--NeedCopy-->
```

To encrypt the cookie by using the configuration utility

1. Navigate to **Traffic Management** > **Change Load Balancing Parameters**, and select **Encode Persistence Cookie Values** and enter a passphrase in **Cookie Passphrase**.

SSL session ID persistence

September 21, 2020

When SSL session ID persistence is configured, the NetScaler appliance uses the SSL session ID, which is part of the SSL handshake process, to create a persistence session before the initial request is directed to a service. The load balancing virtual server directs subsequent requests that have the same SSL session ID to the same service. This type of persistence is used for SSL bridge services.

Note:

There are two issues that users should consider before choosing this type of persistence. First, this type of persistence consumes resources on the NetScaler appliance, which limits the number of concurrent persistence sessions that it can support. If you expect to support a very large number of concurrent persistence sessions, you might want to choose another type of persistence.

Second, if the client and the load-balanced server should renegotiate the session ID during their transactions, persistence is not maintained, and a new persistence session is created when the client's next request is received. This may result in the client's activity on the website being interrupted and the client might be asked to reauthenticate or restart the session. It may also result in large numbers of abandoned sessions if the timeout is set to too large a value.

To configure persistence based on SSL session ID, see Configuring Persistence Types That Do Not Require a Rule..

> **Note**
>
> SSL session ID persistence is not supported with session tickets.

Diameter AVP number persistence

September 21, 2020

You can use persistence based on the Attribute-Value Pair (AVP) number of a Diameter message to create persistent Diameter sessions. When the NetScaler appliance finds the AVP in the Diameter message, it creates a persistence session based on the value of the AVP. All subsequent messages that match the value of the AVP are directed to the previously selected server. If the value of the AVP does not match the persistence session, a new session is created for the new value.

Note: If the AVP number is not defined in Diameter base-protocol RFC 6733, and if the number is nested inside a grouped AVP, you must define a sequence of AVP numbers (maximum of 3) in parent-to-child order. For example, if persist AVP number X is nested inside AVP Y, which is nested in Z, define the list as Z Y X.

To configure Diameter-based persistence on a virtual server by using the command line interface

At the command prompt, type the following command:

```
set lb vserver <name> -PersistenceType <type-> persistAVPno <positive_integer>
<!--NeedCopy-->
```

Example:

```
set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

Custom server ID persistence

September 21, 2020

In the Custom Server ID persistence method, the Server ID specified in the client request is used to maintain persistence. For this type of persistence to work, you must first set a server ID on the services. The NetScaler appliance checks the URL of the client request and connects to the server associated with the specified server ID. The service provider should make sure that the users are aware of the server IDs to be provided in their requests for specific services.

For example, if your site provides different types of data, such as images, text, and multimedia, from different servers, you can assign each server a server ID. On the NetScaler appliance, you specify those server IDs for the corresponding services, and you configure custom server ID persistence on the corresponding load balancing virtual server. When sending a request, the client inserts the server ID into the URL indicating the required type of data.

To configure custom server ID persistence:

- In your load balancing setup, assign a server ID to each service for which you want to use the user-defined server ID to maintain persistence. Alphanumeric server IDs are allowed.

- Specify rules, in the default-syntax expression language, to examine the URL queries for the server ID and forward traffic to the corresponding server.
- Configure custom server ID persistence.

Note: The persistence time-out value does not affect the Custom Server ID persistence type. There is no limit on the maximum number of persistent clients because this persistence type does not store any client information.

Example:

In a load balancing setup with two services, assign server ID 2345-photo-56789 to Service-1, and server ID 2345-drawing-abb123 to Service-2. Bind these services to a virtual server named Web11.

```
set service Service-1 10.102.29.5 -CustomServerID 2345-photo-56789

set service Service-2 10.102.29.6 -CustomServerID 2345-drawing-abb123
<!--NeedCopy-->
```

On virtual server Web11, enable Custom Server ID persistence.

Create the following expression so that all URL queries containing the string “sid=” are examined.

HTTP.REQ.URL.AFTER_STR(“sid=”)

Example:

```
set lb vserver Web11 -persistenceType customserverID -rule "HTTP.REQ.
   URL.AFTER_STR("sid=")"

bind lb vserver Web11 Service-[1-2]
<!--NeedCopy-->
```

When a client sends a request with the following URL to the IP address of Web11, the appliance directs the request to Service-2 and honors persistence.

Example:

`http://www.example.com/index.asp?&sid=2345-drawing-abb123`

For more information about default-syntax policy expressions, see the Policy Configuration and Reference.

To configure custom server ID persistence by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service and set a server ID.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.

4. In Advanced Settings, select Persistence.
5. Select CUSTOMESERVERID, and specify an expression.

IP address persistence

September 21, 2020

You can base persistence on Destination IP addresses, or on both Source IP and Destination IP Addresses.

Persistence Based on Destination IP Addresses

With destination IP address-based persistence, when the NetScaler appliance receives a request from a new client, it creates a persistence session based on the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests to the same destination IP to the same service. This type of persistence is used with link load balancing. For more information about link load balancing, see Link Load Balancing.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address.

To configure persistence based on the destination IP address, see Configuring Persistence Types That Do Not Require a Rule..

Persistence Based on Source and Destination IP Addresses

With source and destination IP address-based persistence, when the NetScaler appliance receives a request, it creates a persistence session based on both the IP address of the client (the source IP address) and the IP address of the service selected by the virtual server (the destination IP address). Subsequently, it directs requests from the same source IP and to the same destination IP to the same service.

The time-out value for destination IP persistence is the same as that for source IP persistence, described in Persistence Based on Source IP Address..

To configure persistence based on both source and destination IP addresses, see Configuring Persistence Types That Do Not Require a Rule..

SIP Call ID persistence

September 21, 2020

With SIP Call ID persistence, the NetScaler appliance chooses a service based on the call ID in the SIP header. This enables it to direct packets for a particular SIP session to the same service and, therefore, to the same load balanced server. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

To configure persistence based on SIP Call ID, see Configuring Persistence Types That Do Not Require a Rule.

RTSP session ID persistence

September 21, 2020

With RTSP Session ID persistence, when the NetScaler appliance receives a request from a new client, it creates a new persistence session based on the Real-Time Streaming Protocol (RTSP) session ID in the RTSP packet header, and then directs the request to the RTSP service selected by the configured load balancing method. It directs subsequent requests that contain the same session ID to the same service. This persistence type is applicable specifically to SIP load balancing. For more information about SIP load balancing, see Monitoring SIP Services.

Note: RTSP Session ID persistence is configured by default on RTSP virtual servers, and you cannot modify that setting.

Sometimes different RTSP servers issue the same session IDs. When this happens, unique sessions cannot be created between the client and the RTSP server by using only the RTSP session ID. If you have multiple RTSP servers that may issue the same session IDs, you can configure the appliance to append the server IP address and port to the session ID, creating a unique token that can be used to establish persistence. This is called session ID mapping.

To configure persistence based on RTSP Session IDs, see Configuring Persistence Types That Do Not Require a Rule.

Important: If you need to use session ID mapping, you must set the following parameter when configuring each service within the load balancing setup. Also, make sure that no non-persistent connections are routed through the RTSP virtual server.

Configure URL passive persistence

September 21, 2020

With URL Passive persistence, when the NetScaler appliance receives a request from a client, it extracts the server IP address-port information (expressed as a single hexadecimal number) from the client request.

URL passive persistence requires configuring an advanced expression that specifies the query element that contains the server IP address-port information. For more information about classic and advanced policy expressions, see Policies and Expressions.

The following expression configures the appliance to examine requests for URL queries that contain the string “urlp=”, extract the server IP address-port information, convert it from a hexadecimal string to an IP and port number, and forward the request to the service configured with this IP address and port number.

HTTP.REQ.URL.AFTER_STR(“urlp=”)

If URL passive persistence is enabled and the above expression is configured, a request with the following URL and server IP address-port string is directed to 10.102.29.10:80.

`http://www.example.com/index.asp?&urlp=0A661D0A0050`

The persistence time-out value does not affect this persistence type; persistence is maintained as long as the server IP address-port information can be extracted from client requests. This persistence type does not consume any appliance resources, so it can accommodate an unlimited number of persistent clients.

To configure URL passive persistence, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule. You set the persistence type to URLPASSIVE. You then perform the procedures provided below.

To configure URL passive persistence by using the CLI

At the command prompt, type:

```
set lb vserver <vserverName> [-persistenceType <persistenceType>] [-rule <expression>]
<!--NeedCopy-->
```

Example:

```
set lb vserver LB-VServer-1 -persistenceType URLPASSIVE – rule HTTP.REQ.URL.AFTER_STR( “urlp=” )
<!--NeedCopy-->
```

To configure persistence on a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advance Settings**, select **Persistence** and specify **URLPASSIVE**.

Configure persistence based on user-defined rules

September 21, 2020

When rule based persistence is configured, the NetScaler appliance creates a persistence session based on the contents of the matched rule before directing the request to the service selected by the configured load balancing method. Subsequently, it directs all requests that match the rule to the same service. You can configure rule based persistence for services of type HTTP, SSL, RADIUS, ANY, TCP, and SSL_TCP.

Rule based persistence requires a classic or default syntax expression. You can use a classic expression to evaluate request headers, or you can use a default syntax expression to evaluate request headers, Web form data in a request, response headers, or response bodies. For example, you could use a classic expression to configure persistence based on the contents of the HTTP Host header. You could also use a default syntax expression to configure persistence based on application session information in a response cookie or custom header. For more information on creating and using classic and default syntax expressions, see Policies and Expressions.

The expressions that you can configure depends on the type of service for which you are configuring rule based persistence. For example, certain RADIUS-specific expressions are not allowed for protocols other than RADIUS, and TCP-option based expressions are not allowed for service types other than the ANY type. For TCP and SSL_TCP service types, you can use expressions that evaluate TCP/IP protocol data, Layer 2 data, TCP options, and TCP payloads.

Note: For a use case that involves configuring rule based persistence on the basis of Financial Information eXchange (“FIX”) Protocol data transmitted over TCP, see
Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream.

Rule based persistence can be used for maintaining persistence with entities such as Branch Repeater appliances, Branch Repeater plug-ins, cache servers, and application servers.

Note: On an ANY virtual server, you cannot configure rule-based persistence for the responses.

To configure persistence based on a user-defined rule, you first configure persistence as described in Configuring Persistence Types That Do Not Require a Rule, and set the persistence type to RULE. You then perform the procedures provided below. You can configure rule based persistence by using the configuration utility or the CLI.

To configure persistence based on user-defined rules by using the CLI

At the command prompt, type:

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression
    >]
<!--NeedCopy-->
```

Example:

```
set lb vserver vsvr_name - rule http.req.header("cookie").value(0).
    typecast_nvlist_t('=',';').value("server")

set lb vserver vsvr_name - resrule http.res.header("set-cookie").value
    (0).typecast_nvlist_t('=',';').value("server")

<!--NeedCopy-->
```

To configure persistence based on user-defined rules by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advanced Settings**, select **Persistence**, select **RULE**, and specify an expression.

Example: Classic Expression for a Request Payload

The following classic expression creates a persistence session based on the presence of a User-Agent HTTP header that contains the string, "MyBrowser", and directs any subsequent client requests that contain this header and string to the same server that was selected for the initial request.

http header User-Agent contains MyBrowser

Example: Default syntax Expression for a Request Header

The following default syntax expression does exactly the same thing as the previous classic expression.

HTTP.REQ.HEADER("User-Agent").CONTAINS ("MyBrowser")

Example: Default syntax Expression for a Response Cookie

The following expression examines responses for "server" cookies, and then directs any requests that contain that cookie to the same server that was selected for the initial request.

HTTP.RES.HEADER("SET-COOKIE").VALUE(0).TYPECAST_NVLIST_T('=';';').VALUE("server")

Configure persistence types that do not require a rule

September 21, 2020

To configure persistence, you must first set up a load balancing virtual server, as described in Setting Up Basic Load Balancing. You then configure persistence on the virtual server.

To configure persistence on a virtual server by using the CLI

At the command prompt, type the following commands to configure persistence and verify the configuration:

```
set lb vserver <name> -PersistenceType <type> [-timeout <integer>]

show lb vserver
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP -timeout 60

show lb vserver
<!--NeedCopy-->
```

Timeout is the time period for which a persistence session is in effect.The timeout default and minimum values (in minutes) vary based on the persistence type as listed in the following table.

persistence type	Default value	Minimum value	Maximum value
Cookie insert/group cookie insert	2	0	1440
Other persistence types	2	2	1440

> **Note**
>
> - Group cookie insert persistence type can be set on the load balancing group.
> - For IP-based persistence, you can also set the persistMask parameter.

> - The persistence type by default is set to NONE.

To configure persistence on a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advanced Settings**, select **Persistence**, and specify a persistence type other than **RULE**.

Configure backup persistence

September 21, 2020

The NetScaler appliance uses backup persistence to choose a new type of persistence when the primary persistence type fails. For example, if the primary persistence type is set to Cookie Insert, and backup persistence is set to Source IP, the NetScaler appliance uses Source IP-based persistence when the cookie is missing from the HT T P header or when the client browser does not support cookies. You can set a time-out value for backup persistence only when the primary persistence type is HTTP Cookie-based or RTSP session ID-based persistence, and the backup persistence type is Source IP-based.

To set backup persistence for a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -persistenceType <PersistenceType> -persistenceBackup <BackupPersistenceType>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -persistenceBackup SourceIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -persistenceBackup SourceIP

set lb vserver Vserver-LB-1  - persistenceType RULE  - rule http.req.header("User-Agent").value(0).contains("MyBrowser")  -persistenceBackup SOURCEIP

set lb vserver Vserver-LB-1 -persistenceType sslsession -persistenceBackup SourceIP
```

```
<!--NeedCopy-->
```

To set backup persistence for a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In **Advanced Settings**, select **Persistence**, and specify a backup persistence type.

Note: The primary persistence must be set to COOKIEINSERT.

Configure persistence groups

September 21, 2020

When you have load-balanced servers that handle several different types of connections (such as Web servers that host multimedia), you can configure a virtual server group to handle these connections. To create a virtual server group, you bind different types of virtual servers, one for each type of connection that your load balanced servers accept, into a single group. You then configure a persistence type for the entire group.

You can configure either source IP-based persistence or HTTP cookie-based persistence for persistence groups. After you set persistence for the entire group, you cannot change it for individual virtual servers in the group. If you configure persistence on a group and then add a new virtual server to the group, the persistence of the new virtual server is changed to match the persistence setting of the group.

When persistence is configured on a group of virtual servers, persistence sessions are created for initial requests, and subsequent requests are directed to the same service as initial request, regardless of the virtual server in the group that receives each client request.

When you add a virtual server that has persistence sessions to a load balancing group with a different persistence type, the existing persistent sessions specific to an old persistence type are deleted. The persistent sessions decide whether the traffic must go to the same virtual server or to a different server. Therefore, existing established connections are not impacted.

The persistence type of a load balancing group is applied to all the virtual servers bound to that group, irrespective of the virtual servers' protocol type. A load balancing group supports the following persistence types:

- SourceIP
- CookieInsert
- Rule

Some virtual servers support only certain persistence types. For example, a virtual server of type SSL_BRIDGE can use only SourceIP persistence type for an LB group.

If you configure HTTP cookie-based persistence, the domain attribute of the HTTP cookie is set. This setting causes the client software to add the HTTP cookie into client requests if different virtual servers have different public host names. For more information about CookieInsert persistence type, see Persistence Based on HTTP Cookies.

To create a virtual server persistency group by using the command line interface

At the command prompt, type:

```
bind lb group <vServerGroupName> <vServerName> -persistenceType <PersistenceType>
<!--NeedCopy-->
```

Example:

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType CookieInsert
<!--NeedCopy-->
```

To modify a virtual server group by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Persistency Groups**, create a persistency group, and specify the virtual servers that must be part of this group.

To modify a virtual server group by using the command line interface

At the command prompt, type:

```
set lb group <vServerGroupName> -PersistenceBackup <BackupPersistenceType> -persistMask <SubnetMaskAddress>
<!--NeedCopy-->
```

Example:

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask 255.255.255.255
<!--NeedCopy-->
```

Configure RADIUS load balancing with persistence

September 21, 2020

Today's complex networking environment often requires coordinating a high-volume, high-capacity load balancing configuration with robust authentication and authorization. Application users may connect to a VPN through mobile access points such as consumer-grade DSL or Cable connections, WiFi, or even dial-up nodes. Those connections usually use dynamic IPs, which can change during the connection.

If you configure RADIUS load balancing on the NetScaler appliance to support persistent client connections to RADIUS authentication servers, the appliance uses the user logon or the specified RADIUS attribute instead of the client IP as the session ID, directing all connections and records associated with that user session to the same RADIUS server. Users are therefore able to log on to your VPN from mobile access locations without experiencing disconnections when the client IP or WiFi access point changes.

To configure RADIUS load balancing with persistence, you must first configure RADIUS authentication for your VPN. For information and instructions, see the Authentication, Authorization, Auditing (AAA) chapter in AAA Application Traffic. You must also choose either the Load Balancing or Content Switching feature as the basis for your configuration, and make sure that the feature you chose is enabled. The configuration process with either feature is almost the same.

Then, you configure either two load balancing, or two content switching, virtual servers, one to handle RADIUS authentication traffic and the other to handle RADIUS accounting traffic. Next, you configure two services, one for each load balancing virtual server, and bind each load balancing virtual server to its service. Finally, you create a load balancing persistency group and set the persistency type to RULE.

To configure RADIUS load balancing with persistence, see the following sections:

- Enabling the Load Balancing or Content Switching Feature
- Configuring Virtual Servers
- Configuring Services
- Binding Virtual Servers to Services
- Configuring a Persistency Group for Radius

Enabling the Load Balancing or Content Switching Feature

To use the Load Balancing or Content Switching feature, you must first ensure that the feature is enabled. If you are configuring a new NetScaler appliance that has not previously been configured, both of these features are already enabled, so you can skip to the next section. If you are configuring a

NetScaler appliance with a previous configuration on it, and you are not certain that the feature you will use is enabled, you must do that now.

- For instructions on enabling the load balancing feature, see Enabling Load Balancing.
- For instructions on enabling the content switching feature, see Enabling Content Switching.

Configuring Virtual Servers

After enabling the load balancing or content switching feature, you must next configure two virtual servers to support RADIUS authentication:

- **RADIUS authentication virtual server.** This virtual server and its associated service will handle authentication traffic to your RADIUS server. Authentication traffic consists of connections associated with users logging onto your protected application or virtual private network (VPN).
- **RADIUS accounting virtual server.** This virtual server and its associated service will handle accounting connections to your RADIUS server. Accounting traffic consists of connections that track an authenticated user's activities on your protected application or VPN.

Important: You must create either a pair of load balancing virtual servers or a pair of content switching virtual servers to use in your RADIUS persistence configuration. You cannot mix virtual server types.

To configure a load balancing virtual server by using the command line interface

At the command prompt type the following commands to create a new load balancing virtual server and verify the configuration:

```
add lb vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule <rule>

show lb vserver <name>
<!--NeedCopy-->
```

To configure an existing load balancing virtual server, replace the above add lb virtual server command with the set lb vserver command, which takes the same arguments.

To configure a content switching virtual server by using the command line interface

At the command prompt type the following commands to create a new content switching virtual server and verify the configuration:

```
add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule <rule>

show cs vserver <name>
```

```
<!--NeedCopy-->
```

To configure an existing content switching virtual server, replace the above add cs vserver command with the set cs vserver command, which takes the same arguments.

Example:

```
add lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

add lb vserver radius_acct_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.33 1812 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME

set lb vserver radius_auth_vs1 RADIUS 192.168.46.34 1813 -lbmethod
    TOKEN -rule CLIENT.UDP.RADIUS.USERNAME
<!--NeedCopy-->
```

To configure a load balancing or content switching virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers or navigate to Traffic Management > Content Switching > Virtual Servers», and configure a virtual server.

Configuring Services

After configuring your virtual servers, you must next configure two services, one for each of the virtual servers that you created.

Note: Once configured, these services are in the DISABLED state until the NetScaler appliance can connect to your RADIUS server's authentication and accounting IPs and monitor their status. For instructions, see Configuring Services.

Binding Virtual Servers to Services

After configuring your services, you must next bind each of the virtual servers that you created to the appropriate service.For instructions, see Binding Services to the Virtual Server.

Configuring a Persistency Group for Radius

After binding your load balancing virtual servers to the corresponding services, you must set up your RADIUS load balancing configuration to support persistence. To do so, you configure a load balancing persistency group that contains your RADIUS load balancing virtual servers and services, and configure that load balancing persistency group to use rule-based persistence. A persistency group is required because the authentication and accounting virtual servers are different and both the authentication & accounting message for a single user should reach the same RADIUS server. By configuring a persistency group, the same session is used for both virtual servers. For instructions, see Configuring Persistence Groups.

View persistence sessions

September 21, 2020

You can view the different persistence sessions that are in effect globally or for a particular virtual server.

Note: A NetScaler nCore appliance uses multiple CPU cores for packet handling. Every session on the appliance is owned by a CPU core. If the appliance receives a request for which a session does not already exist, a session is created, and one of the cores is designated as the owner of that session. Subsequent requests that belong to that session might not always arrive at and be handled by the owner core. In that case, inter-core messaging ensures that the session information on the owner core is always current. However, when a core receives a request that belongs to a persistence session owned by another core, the inter-core messaging does not refresh the timeout value for the persistence session. Consequently, in the output of successively executed show lb persistentSessions commands, which display timeout values from owner cores only, the timeout value for a persistence session might diminish to 0 (zero), even if the persistence session continues to be active.

To view a persistence session by using the command line interface

At the command prompt, to view all persistence sessions type:

```
show lb persistentSessions [<vServer>]
<!--NeedCopy-->
```

Example:

```
show lb persistentSessions myVserver
<!--NeedCopy-->
```

To view persistence sessions by using the configuration utility

Navigate to Traffic Management > Virtual server persistent sessions.

Clear persistence sessions

September 21, 2020

You might need to clear persistence sessions from the NetScaler appliance if sessions fail to time out. You can do one of the following:

- Clear all sessions for all virtual servers at once.
- Clear all sessions for a given virtual server at once.
- Clear a particular session that is associated with a given virtual server.

To clear a persistence session by using the command line interface

At the command prompt, type the following commands to clear persistence sessions and verify the configuration:

```
clear lb persistentSessions [<vServer> [-persistenceParam <string>]]

show persistentSessions <vServer>
<!--NeedCopy-->
```

Examples:

Example 1 clears all persistence sessions for load balancing virtual server lbvip1.

Example 2 first displays the persistence sessions for load balancing virtual server lbvip1, clears the session with persistence parameter xls, and then displays the persistence sessions to verify that the session was cleared.

Example 1:

```
> clear persistentSessions lbvip1
 Done
> show persistentSessions
 Done
>
<!--NeedCopy-->
```

Example 2:

```
> show persistentSessions lbvip1
Type         SRC-IP     ...    PERSISTENCE-PARAMETER
RULE         0.0.0.0    ...    xls
RULE         0.0.0.0    ...    txt
RULE         0.0.0.0    ...    html
 Done
> clear persistentSessions lbvip1 -persistenceParam xls
 Done
> show persistentSessions lbvip1
Type         SRC-IP     ...    PERSISTENCE-PARAMETER
RULE         0.0.0.0    ...    txt
RULE         0.0.0.0    ...    html
 Done
>
<!--NeedCopy-->
```

To clear persistence sessions by using the configuration utility

1. Navigate to Traffic Management > Clear Persistent Sessions.

Override persistence settings for overloaded services

September 21, 2020

When a service is loaded or is otherwise unavailable, service to clients is degraded. To work around this situation, you might have to configure the NetScaler appliance to temporarily forward to other services the requests that would otherwise be included in the persistence session that is associated with the overloaded service. In other words, you might have to override the persistence setting that is configured for the load balancing virtual server. You can achieve this functionality by setting the skip-persistency parameter. With the parameter set, when the virtual server receives new connections for an overloaded service, the virtual server disregards any existing persistence sessions that are associated with that service, until the service returns to a state at which it can accept requests. Persistence sessions associated with other services are not affected. The functionality is available for only virtual servers whose type is ANY or UDP.

In Branch Repeater load balancing configurations, you must also configure a load monitor and bind it to the service. The monitor takes the service out of subsequent load balancing decisions until the load on the service is brought below the configured threshold. For information about configuring a load monitor for your virtual server, see Understanding Load Monitors.

You can configure the virtual server to perform one of the following actions with the requests that would otherwise form a part of the persistence session:

- **Send each request to one of the other services.** The virtual server takes a load balancing decision and sends each request to one of the other services on the basis of the configured load balancing method. If all the services are overloaded, requests are dropped until a service becomes available.

  Both wildcard and IP address–based virtual servers support this option. This action is appropriate for all deployments, including deployments in which the virtual server is load balancing Branch Repeater appliances or firewalls.

- **Bypass the virtual server-service configuration.** The virtual server does not take a load balancing decision. Instead, it simply bridges each request through to a physical server on the basis of the destination IP address in the request.

  Only wildcard virtual servers of type ANY and UDP support the bypass option. Wildcard virtual servers have a *:* IP and port combination. This action is appropriate for deployments in which you are using the virtual server to load balance Branch Repeater appliances or firewalls. In these deployments, the NetScaler appliance first forwards a request to a Branch Repeater appliance or firewall, and then forwards the processed response to a physical server. If you configure the virtual server to bypass the virtual server–service configuration for overloaded services, if a Branch Repeater appliance or firewall gets overloaded, the virtual server bridges requests directly to their destination IP addresses until the Branch Repeater appliance or firewall can accept requests.

To override persistence settings for overloaded services by using the command line interface

At the command prompt, type the following commands to override persistence settings for overloaded services and verify the configuration:

```
set lb vserver <name> -skippersistency <skippersistency>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver mylbvserver -skippersistency ReLb
 Done
> show lb vserver mylbvserver
    mylbvserver (*:*) - ANY Type: ADDRESS
```

```
        . . .
        . . .
 Skip Persistency: ReLb
        . . .
 Done
>
<!--NeedCopy-->
```

To override persistence settings for overloaded services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers and select the virtual server of type UDP or ANY.
2. In the Advanced Settings pane, select Traffic Settings, and specify the type of Skip Persistency.

Troubleshooting

September 21, 2020

The statistics from the NetScaler VPX appliance indicate that the appliance has reached the session persistence limit. As a result, persistence sessions are failing. Is possible to increase the session persistence limit?

Cause: The NetScaler appliance has the system limit of 250,000 persistence session for a core.

Resolution: To resolve this issue, you can perform any of the following tasks:

- Reduce the time out value for persistence
- Increase the number of cores for the appliance

After configuring Cookie Insert persistence on the NetScaler appliance, the users report that the connections work fine for some time, but then start getting disconnected. What best practice should I follow when configuring persistence?

Cause: By default, the time-out value for Cookie Insert persistence is 120 seconds.

Resolution: When you configure persistence for applications for which idle time cannot be determined, set the Cookie Insert persistence time-out value to 0. With this setting, the connection does not time out.

After configuring an HTTP virtual server on the NetScaler appliance, I need to make sure that a user always connects to the same server for the requested content, so I configured SourceIP persistence. Now, increasing the time-out value for persistence introduces latency. How can I increase the timeout value without affecting performance?

Resolution: Consider using Cookie Insert persistence with the time-out value set to 0. This setting enables long-duration persistence settings, because the appliance does not specify a time for expiring the cookie.

After configuring Cookie Insert persistence on the NetScaler appliance, it works as expected when clients from the same time zone access the content. However, when a client from another time zone makes an attempt to connect, the connection is immediately timed out.

Cause: Time based Cookie Insert persistence works as expected when a client from the same time zone makes a connection. However, when the client machine and NetScaler appliance are in different time zones, the cookie is not valid. For example, when a client in EST time zone sends a cookie at 11:00 AM EST to a NetScaler appliance in the PST time zone, the appliance receives the cookie at 2:00 PM PST. As a result of the difference in time, the cookie is not valid, and the connection is immediately timed-out.

Resolution: Set the time-out value for Cookie Insert persistence to 0.

A NetScaler appliance is used to load balance application servers, such as Oracle Weblogic server. To make sure that clients get persistent connections to these servers, SourceIP persistence is configured. It works as expected when a connection is made from a computer. However, when thin clients attempt a connection through a terminal server and, as a result, the appliance receives requests from multiple clients from the same IP address (the terminal server IP address). Therefore, the connections from all thin clients are directed to the same application server. Is it possible to configure persistency for requests from individual thin clients based on the client IP address?

Cause: The NetScaler appliance receives requests from the terminal server and the source IP address of the request remains the same. As a result, the appliance cannot distinguish among the requests received from the thin clients and provide persistence according to the requests from thin clients.

Resolution: To avoid this problem, you can configure Rule persistence based on some unique parameter value for each thin client.

The NetScaler appliance is used to load balance Web Interface servers. When accessing the servers, the user receives the "State Error" error message. Additionally, when one of the Web Interface servers is shut down or not available, some of the users receive an error message.

Cause: Lack of persistence to the Web Interface servers can result in error messages when a user attempts to connect to the server.

Resolution: Citrix recommends that you specify the Cookie Insert persistence method on the NetScaler appliance when load balancing Web Interface servers.

Insert cookie attributes to ADC generated cookies

September 21, 2020

The web administrators can insert additional cookie attributes to the cookies generated by Netscaler appliance. These additional cookie attributes help in enforcing the required policies for the ADC generated cookies based on the application access pattern.

The following features use the ADC generated cookies to achieve persistency.

- Load balancing cookie persistence
- Load balancing group cookie persistence
- GSLB site persistence

You can insert additional cookie attributes to the ADC generated cookies using the following parameters:

- **LiteralADCCookieAttribute:** Append additional cookie attributes to the ADC generated cookie, as a string.
- **ComputedADCCookieAttribute:** Use an ADC ns variable to conditionally append cookie attributes to the ADC generated cookie, based on the client or server attributes, for example, user agent version.

> **Note**
>
> You cannot configure both Literal ADC Cookie Attribute and Computed ADC Cookie Attribute, simultaneously on the load balancing parameter or in a single load balancing profile.

Use case: Configure SameSite cookie attribute

Every cookie has a domain associated with it. When a cookie's domain matches the website domain in the user's address bar, this is considered a same-site (or first party) context. If the domain associated with a cookie matches an external service and not the website in the user's address bar, this is considered a cross-site (or third party) context.

The SameSite attribute indicates the browser whether the cookie can be used for cross-site context or only for same-site context. Also, if an application intends to be accessed in cross-site context then it can do so only via HTTPS connection. For details, see RFC6265.

Until Feb 2020, the SameSite property was not explicitly set in Netscaler. The browser took the default value as None, and did not impact the Netscaler deployments.

However with certain browsers upgrade, such as Google Chrome 80, there is a change in the default cross-domain behavior of cookies. The SameSite attribute can be set to one of the following values. Default value for Google Chrome is set to Lax.

- **None:** Indicates the browser to use cookie in cross-site context only on secure connections.
- **Lax:** Indicates the browser to use cookie for requests in the same-site context. In cross-site context, only safe HTTP methods like GET request can use the cookie.
- **Strict:** Use the cookie only in the same-site context.
  If there is no SameSite attribute in the cookie, the Google Chrome assumes the functionality of SameSite=Lax.

> **Note**
>
> For certain version of other browsers, the default value for SameSite attribute might be set to **None**. In some browser versions, "SameSite = none" can be treated differently. For example, the following browsers reject a cookie with "SameSite = none":
>
> - Versions of Chrome from Chrome 51 to Chrome 66 (inclusive on both ends)
> - Versions of UC browser on Android prior to version 12.13.2

Configure ADC generated cookies

To configure ADC generated cookie attributes, you must perform the following:

1. Create a load balancing virtual server
2. Set the ADC Cookie attributes for the load balancing virtual server, either through LB parameters or LB profile.
3. If you use an LB profile, set the LB profile to a load balancing virtual server.
4. If you choose to use the Computed ADC Cookie Attribute, configure the related rewrite policy.

> **Note**
>
> If an LB profile is bound to an LB virtual server, then the profile parameter configuration is considered instead of global LB parameter configuration.

You can set the ADC generated cookie attributes by the following methods:

- Setting the ADC cookie attributes in LB parameters
- Setting the ADC cookie attributes in LB profile

Setting the ADC cookie attributes in LB parameters by using the CLI

To apply a policy uniformly to ADC generated cookies of all applications configured on Netscaler appliance, you can set the ADC cookie attribute in the global LB parameters.

The **Literal ADC Cookie Attribute** setting allows you to unconditionally insert the cookie attributes to the ADC generated cookie.

At the command prompt, type:

```
set lb parameter -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

Example:

```
set lb parameter -LiteralADCCookieAttribute SameSite=None
<!--NeedCopy-->
```

The **Computed ADC Cookie Attribute** setting allows you to conditionally insert the cookie attributes, based on the client or server attributes, to the ADC generated cookie.

At the command prompt, type:

```
set lb parameter -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

Example:

```
add ns variable cookieattribute_var -type "text(100)" -scope transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set "\"SameSite=None\""

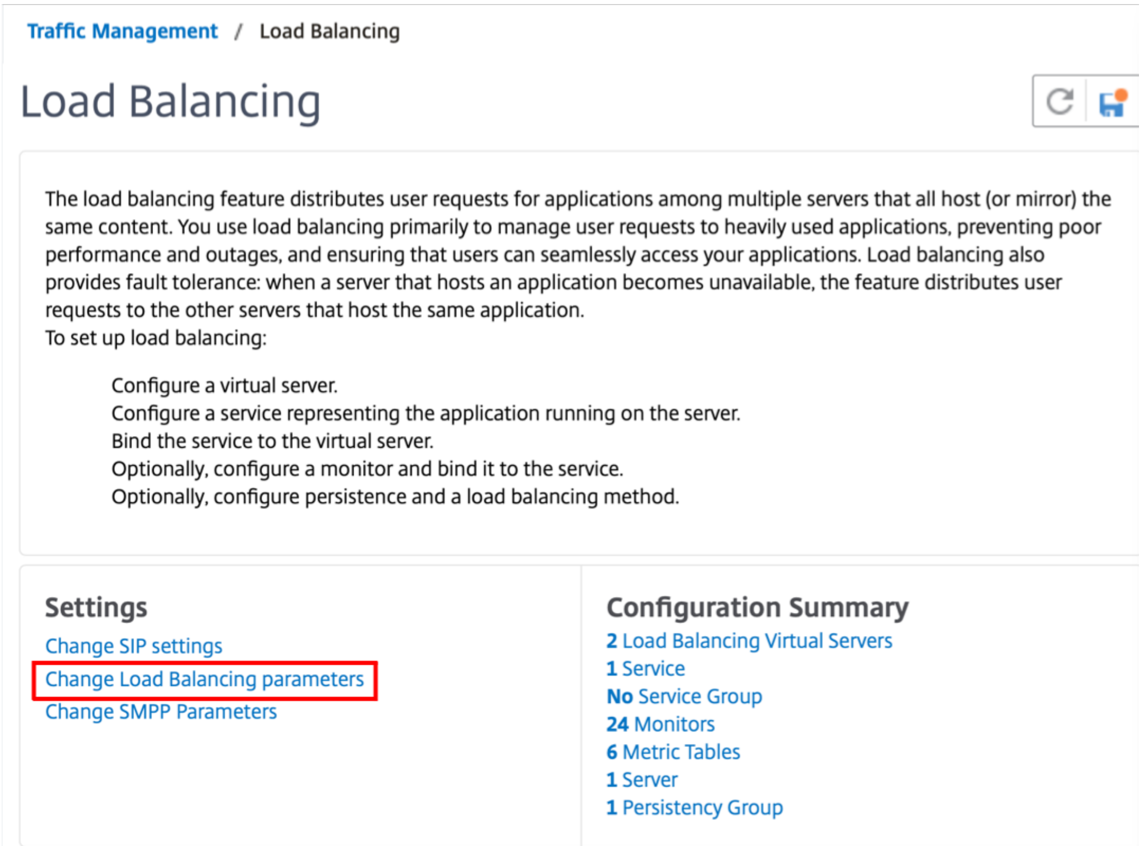
add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re/OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_insert "pol_iphone || pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

bind rewrite global exception_samesite_attribute 90 110 -type RES_OVERRIDE
bind rewrite global append_samesite_attribute 100 110 -type RES_OVERRIDE

<!--NeedCopy-->
```

Setting the ADC cookie attributes in LB parameters by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Change Load Balancing parameters**.



Traffic Management / Load Balancing

Load Balancing

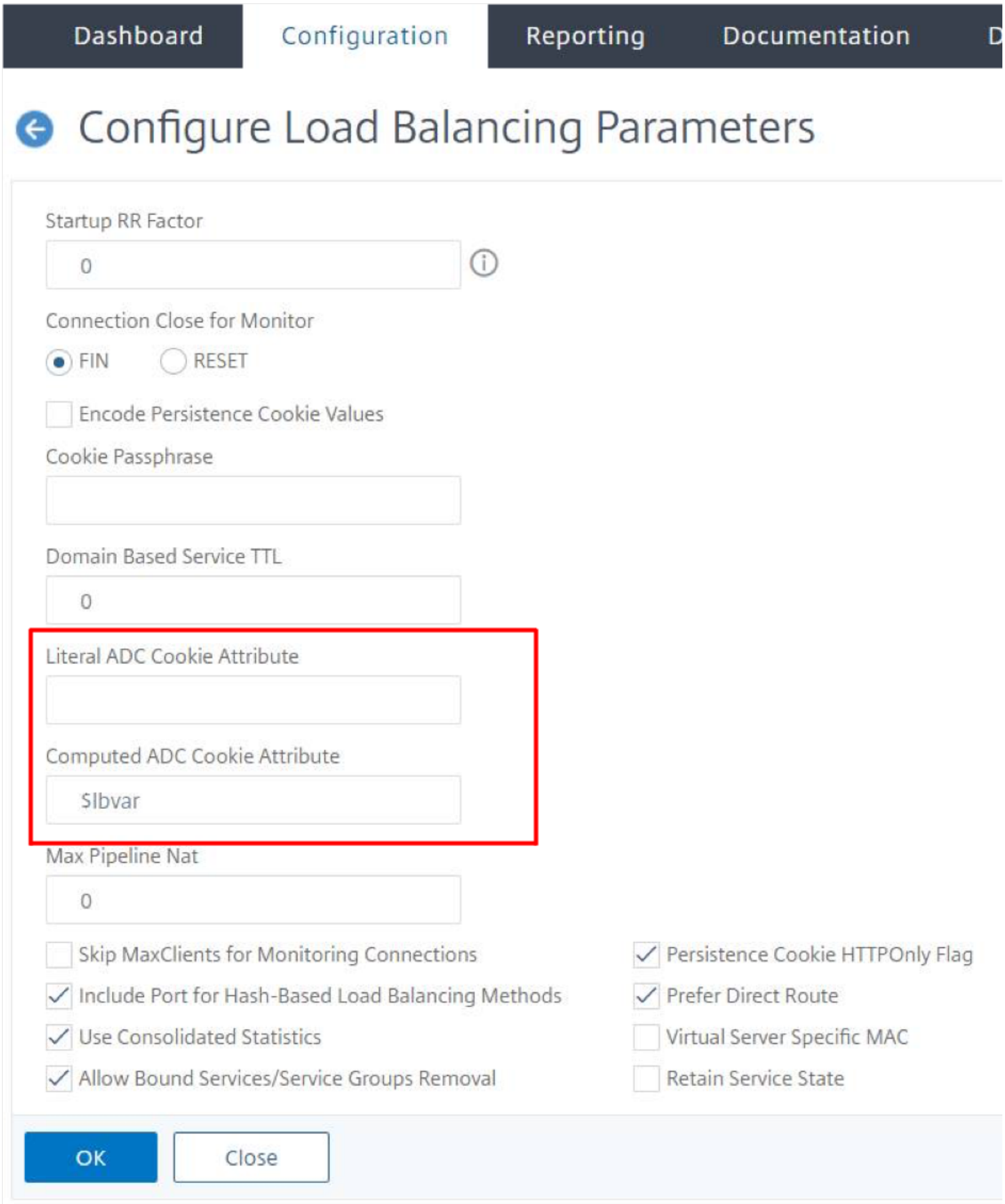
The load balancing feature distributes user requests for applications among multiple servers that all host (or mirror) the same content. You use load balancing primarily to manage user requests to heavily used applications, preventing poor performance and outages, and ensuring that users can seamlessly access your applications. Load balancing also provides fault tolerance: when a server that hosts an application becomes unavailable, the feature distributes user requests to the other servers that host the same application.
To set up load balancing:

Configure a virtual server.
Configure a service representing the application running on the server.
Bind the service to the virtual server.
Optionally, configure a monitor and bind it to the service.
Optionally, configure persistence and a load balancing method.

Settings
Change SIP settings
Change Load Balancing parameters
Change SMPP Parameters
Configuration Summary
2 Load Balancing Virtual Servers
1 Service
No Service Group
24 Monitors
6 Metric Tables
1 Server
1 Persistency Group

2. In the **Configure Load Balancing Parameters** pane, enter appropriate values for either one of the fields based on your requirement:

   - **Literal ADC Cookie Attribute**
   - **Computed ADC Cookie Attribute**



Configure Load Balancing Parameters

Startup RR Factor: 0

Connection Close for Monitor: ● FIN ○ RESET

☐ Encode Persistence Cookie Values

Cookie Passphrase:

Domain Based Service TTL: 0

Literal ADC Cookie Attribute:

Computed ADC Cookie Attribute: $lbvar

Max Pipeline Nat: 0

☐ Skip MaxClients for Monitoring Connections
☑ Include Port for Hash-Based Load Balancing Methods
☑ Use Consolidated Statistics
☑ Allow Bound Services/Service Groups Removal
☑ Persistence Cookie HTTPOnly Flag
☑ Prefer Direct Route
☐ Virtual Server Specific MAC
☐ Retain Service State

3. Click **OK**.

Setting the ADC cookie attributes in LB profile by using the CLI

To apply a policy for a specific application that is configured on Netscaler appliance, you can set the cookie attribute parameters in the LB profile bound to the application-specific LB virtual server.

The **Literal ADC Cookie Attribute** setting in the LB profile allows you to unconditionally insert the cookie attributes to the ADC generated cookie that is specific to a virtual server.

At the command prompt, type:

```
add lb profile <profile name> -LiteralADCCookieAttribute <string>
<!--NeedCopy-->
```

Example:

```
add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
   =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
   COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
<!--NeedCopy-->
```

The **Computed ADC Cookie Attribute** setting in the LB profile allows you to conditionally insert the cookie attributes based on the client or server attributes, to the ADC generated cookie. Then, set this LB profile to an LB virtual server.

At the command prompt, type:

```
add lb profile <profile name> -ComputedADCCookieAttribute <ns variable>
<!--NeedCopy-->
```

Example:

```
add ns variable cookieattribute_var -type "text(100)" -scope
   transaction
add ns assignment samesiteassign -variable "$cookieattribute_var" -set
   "\"SameSite=None\""
add lb profile LB-Vserver-Profile-1 -ComputedADCCookieAttributE "
   $cookieattribute_var"

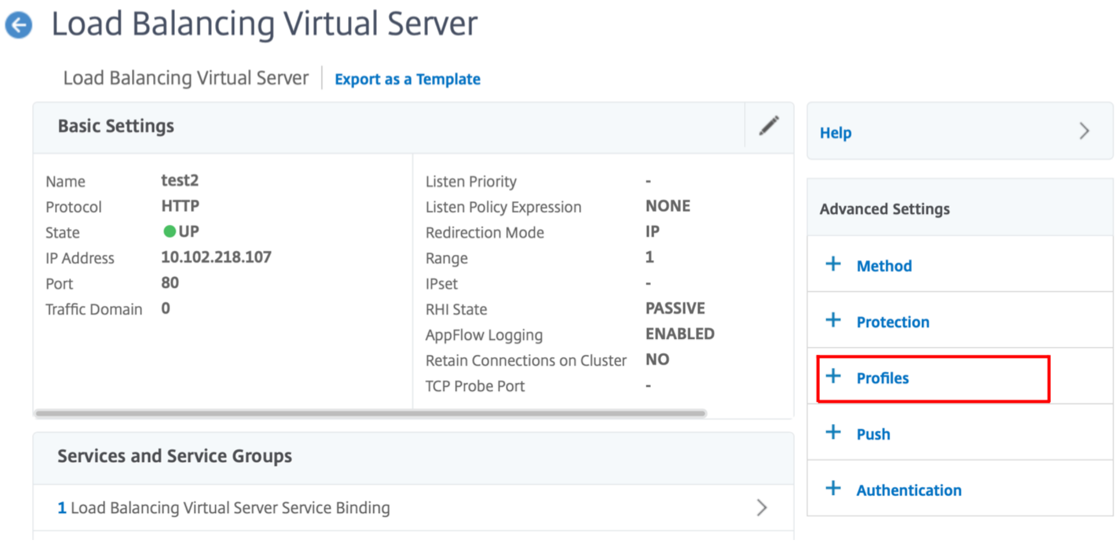
add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").
   CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re
   /OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).
   typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").
   CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT
   (re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).
   BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_attribute "pol_iphone ||
   pol_chrome " NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
   COOKIEINSERT -lbprofilename LB-Vserver-Profile-1
```

```
bind lb vserver LB-VServer-1 -policyName exception_samesite_attribute -priority 90 -gotoPriorityExpression 110 -type RESPONSE
bind lb vserver LB-VServer-1 -policyName append_samesite_attribute -priority 100 -gotoPriorityExpression 110 -type RESPONSE
<!--NeedCopy-->
```

Setting the ADC Cookie attributes in LB profile by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual servers**.
2. Select a virtual server and click **Edit**.
3. Under **Advanced Settings** section, click **Add profiles**.



Load Balancing Virtual Server

Basic Settings

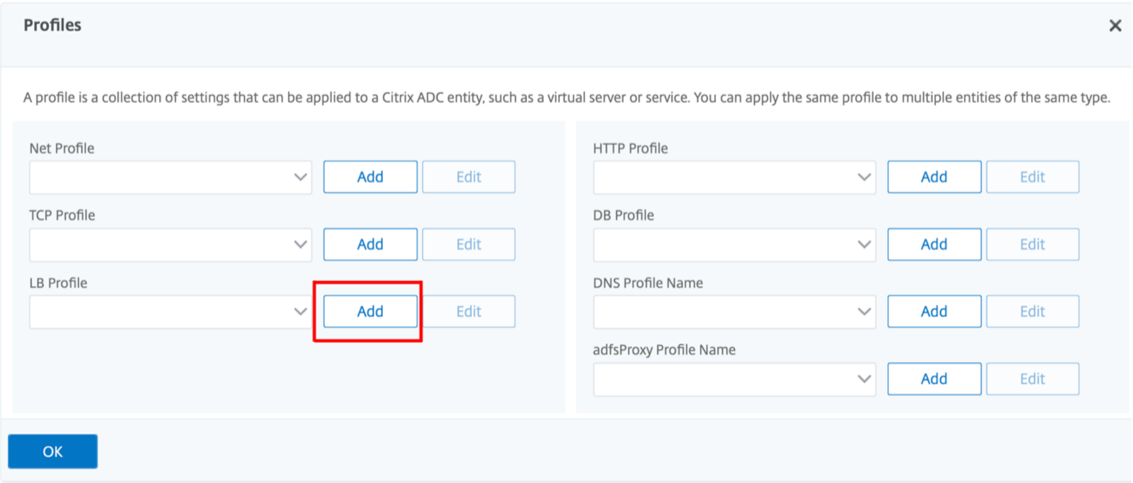
Name test2
Protocol HTTP
State UP
IP Address 10.102.218.107
Port 80
Traffic Domain 0
Listen Priority -
Listen Policy Expression NONE
Redirection Mode IP
Range 1
IPset -
RHI State PASSIVE
AppFlow Logging ENABLED
Retain Connections on Cluster NO
TCP Probe Port -

Services and Service Groups

1 Load Balancing Virtual Server Service Binding

Load Balancing Virtual Server
Export as a Template
Help
Advanced Settings
+ Method
+ Protection
+ Profiles
+ Push
+ Authentication

4. In the **Profiles** section, click **Add** to create an LB Profile.
   If you have already created a profile, choose it from the **LB Profile** drop-down menu.



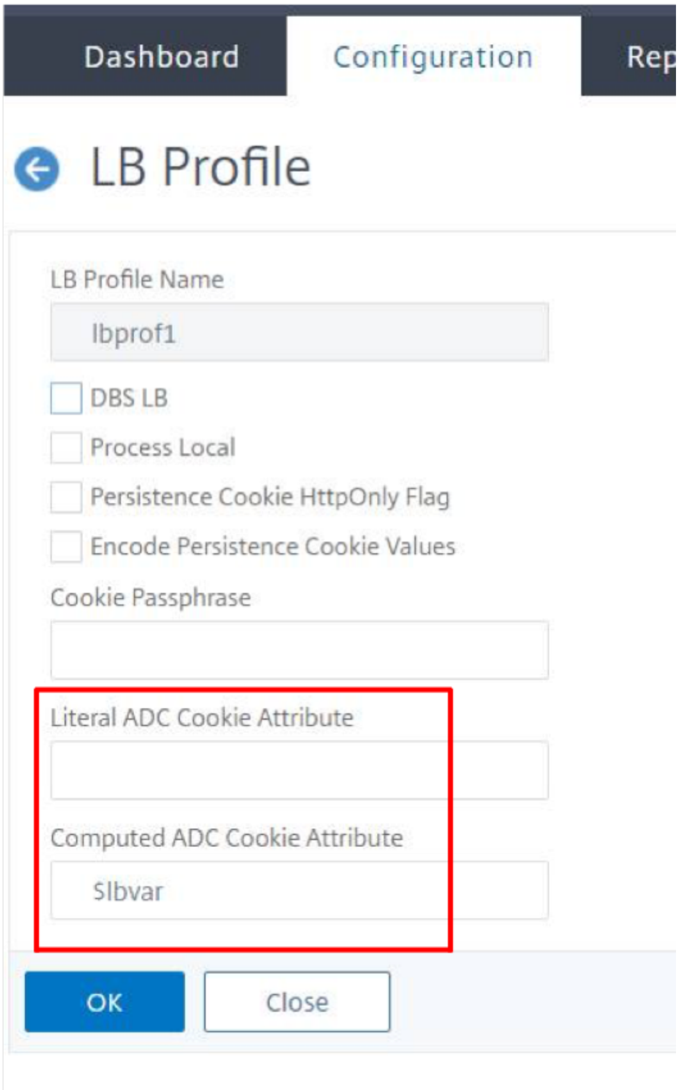
Profiles

A profile is a collection of settings that can be applied to a Citrix ADC entity, such as a virtual server or service. You can apply the same profile to multiple entities of the same type.

Net Profile
TCP Profile
LB Profile
HTTP Profile
DB Profile
DNS Profile Name
adfsProxy Profile Name
Add
Edit
OK

5. In the **LB Profile** pane, enter appropriate values for either one of the fields based on your requirement:

- **Literal ADC Cookie Attribute**
- **Computed ADC Cookie Attribute**



LB Profile Name: lbprof1

☐ DBS LB
☐ Process Local
☐ Persistence Cookie HttpOnly Flag
☐ Encode Persistence Cookie Values

Cookie Passphrase:

Literal ADC Cookie Attribute:

Computed ADC Cookie Attribute: $lbvar

1. Click **OK**.
2. Set the created LB profile to the LB virtual server created in **Step 1**.

Verify ns variable configuration

To verify that the ADC ns variable is configured appropriately in LB parameters or LB profile, use the show lb parameter or show lb profile commands.

The following table lists the various warning messages and its cause, when the ns variable is not correctly configured.

Warning message	Reasons
NS Variable is not configured. Configure it with type text() and scope transaction for variable	NS variable is not yet configured.
Scope of configured NS variable is not transaction.	Variable is configured but scope is not set to “transaction.”
Type of variable is not Text().	Variable is configured but type is not set to “Text”.
Configured value-max-size for NS Variable is greater than 255.	The value configured for NS variable is greater than 255 characters. **Note:** A maximum length of 255 characters can be appended to an ADC generated cookie. The characters that exceed the maximum length are truncated.

Sample output

In the following example, the warning message is displayed when the ns variable is not configured.

```
set lb parameter -ComputedADCCookieAttribute "$lbvar"

Warning: NS Variable is not configured. Please configure it with type text() and scope transaction
Done
<!--NeedCopy-->
```

The warning message is displayed in the following output of the `show lb parameter` command.

```
show lb parameter

Global LB parameters:
Persistence Cookie HttpOnly Flag: ENABLED
Use Encrypted Persistence Cookie: DISABLED
Use Port For Hash LB: YES
Prefer direct route: YES
Retain Service State: OFF
Start RR Factor: 0
Skip Maxclient for Monitoring: DISABLED
Monitor Connection Close: FIN
Use consolidated stats for LeastConnection: YES
```

```
Allow mac mode based vserver to pick thereturn traffic from services:
    DISABLED
Allow bound service removal: ENABLED
TTL for Domain Based Server: 0 secs

Citrix ADC Cookie Variable Name: $lbvar(NS Variable is not configured.
    Please configure it with type text() and scope transaction)

Done
<!--NeedCopy-->
```

Sample configuration for inserting cookie attributes in GSLB deployment

The following sample configuration applies to site persistence configured on GSLB services corresponding to an LB virtual server. To append some additional cookie attributes to the GSLB cookies, perform the following configuration.

- Set the ADC cookie attributes in the LB profile (LB-Vserver-Profile-1).
- Set the Literal ADC Cookie Attribute value, for example "SameSite=None", in the LB profile.
- Set the LB profile to the load balancing virtual server (LB-VServer-1), which represents the GSLB service.

```
add gslb vserver GSLB-VServer-1 SSL -backupLBMethod ROUNDROBIN -
    tolerance 0 -appflowLog DISABLED
add gslb site site1 10.102.148.4 -publicIP 10.102.148.4
add gslb service site1_gsvc1 10.102.148.35 SSL 443 -publicIP
    10.102.148.35 -publicPort 443 -maxClient 0 -siteName site1 -
    sitePersistence HTTPRedirect -sitePrefix ss1 -cltTimeout 180 -
    svrTimeout 360 -downStateFlush ENABLED

bind gslb vserver GSLB-VServer-1 -serviceName site1_gsvc1
bind gslb vserver GSLB-VServer-1 -domainName www.gslb.com -TTL 5

add service service-1 10.102.84.140 SSL 443

add lb profile LB-Vserver-Profile-1 -LiteralADCCookieAttribute SameSite
    =None
add lb vserver LB-VServer-1 SSL 10.102.148.37 443 -persistenceType
    COOKIEINSERT -lbprofilename LB-Vserver-Profile-1

bind lb vserver LB-VServer-1 service-1
<!--NeedCopy-->
```

> **Note**
>
> You can also conditionally insert the cookie attributes using the Computed ADC Cookie Attribute.

Sample configuration for inserting cookie attribute in content switching deployment

The following sample configuration applies when multiple applications are hosted behind a content switching virtual server. To apply the same policy to all the applications, bind the rewrite policies to content switching virtual server instead of LB virtual server, as follows:

- Set the ADC cookie attributes in the LB parameters.

  > **Note:**
  >
  > You can set the ADC cookie attributes in the LB profile as well.

- Configure the ns variable (cookieattribute_var) of Type set to Text and Scope set to Transaction.
- Set the Computed ADC Cookie Attribute in the global LB parameters using the ns variable.
- Set the rewrite policies (exception_samesite_attribute and append_samesite_attribute) to the content switching virtual servers for inserting the cookie attributes.

```
add ns variable cookieattribute_var -type "text(100)" -scope transaction
set lb parameter -ComputedADCCookieAttributE "$cookieattribute_var"
add ns assignment samesiteassign -variable "$cookieattribute_var" -set "\"SameSite=None\""

add policy expression pol_iphone "(HTTP.REQ.HEADER(\"User-Agent\").CONTAINS(\"iP\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re/OS \\d+\\_/).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).EQ(12).typecast_text_t ALT \"false\").eq(\"true\"))"
add policy expression pol_chrome "(HTTP.REQ.HEADER(\"User-Agent\").CONTAINS(\"Chrom\") && (HTTP.REQ.HEADER(\"User-Agent\").REGEX_SELECT(re/Chrom.*\\d+./).REGEX_SELECT(re/\\d+/).TYPECAST_NUM_T(DECIMAL).BETWEEN(51,66).typecast_text_t ALT \"false\").eq(\"true\"))"
add rewrite policy exception_samesite_insert "pol_iphone || pol_chrome" NOREWRITE
add rewrite policy append_samesite_attribute true samesiteassign

add lb vserver LB-VServer-1 SSL 10.102.148.35 443
add lb vserver LB-VServer-2 SSL 10.102.148.36 443

add cs vserver CS-VServer-1 SSL 10.102.148.42 443 -persistenceType COOKIEINSERT

add cs action act1 -targetLBVserver v1
add cs action act2 -targetLBVserver v2
add cs policy CS-policy-1 -rule "HTTP.REQ.URL.CONTAINS(\"file1.html\")"
     -action act1
add cs policy CS-policy-2 -rule "HTTP.REQ.URL.CONTAINS(\"file2.html\")"
     -action act2

bind cs vserver CS-VServer-1 -policyName CS-policy-1 -priority 1
bind cs vserver CS-VServer-1 -policyName CS-policy-2 -priority 2

bind cs vserver -policyname exception_samesite_attribute 90 110 -type
    RES_OVERRIDE
bind cs vserver -policyname append_samesite_attribute 100 110 -type
    RES_OVERRIDE
<!--NeedCopy-->
```

Customize a load balancing configuration

September 21, 2020

After you configure a basic load balancing setup, you can make a number of modifications to it so that it distributes load exactly as you need. The load balancing feature is complex. You can modify the basic elements by changing the load balancing algorithm, configuring load balancing groups and using them to create your load balancing configuration, configuring persistent client-server connections, configuring the redirection mode, and assigning different weights to different services that have different capacities.

The default load balancing algorithm on the NetScaler appliance is the least connection method, which configures the appliance to send each incoming connection to the service that is currently handling the fewest connections. You can specify different load balancing algorithms, each of which is suited to different conditions.

To accommodate applications such as shopping carts, which require that all requests from the same user be directed to the same server, you can configure the appliance to maintain persistent connections between clients and servers. You can also specify persistence for a group of virtual servers, causing the appliance to direct individual client requests to the same service regardless of which virtual server in the group receives the client request.

You can enable and configure the redirection mode that the appliance uses when redirecting user requests, choosing between IP-based and MAC-based forwarding. You can assign weights to different services, specifying what percentage of incoming load should be directed to each service, so that you can include servers with different capacities in the same load balancing setup without overloading the lower-capacity servers or allowing the higher-capacity servers to sit idle.

This section includes the following details:

- Customizing the Hash Algorithm for Persistence across Virtual Servers
- Configuring the Redirection Mode
- Configuring per-VLAN Wildcarded Virtual Servers
- Assigning Weights to Services
- Configuring the MySQL and Microsoft SQL Server Version Setting

Customize the hash algorithm for persistence across virtual servers

September 21, 2020

The NetScaler appliance uses hash-based algorithms for maintaining persistence across virtual servers. By default, the hash-based load balancing method uses a hash value of the IP address and port number of the service. If a service is made available at different ports on the same server, the algorithm generates different hash values. Therefore, different load balancing virtual servers might send requests for the same application to different services, breaking the pseudo-persistence.

As an alternative to using the port number to generate the hash value, you can specify a unique hash identifier for each service. For a service, the same hash identifier value must be specified on all the virtual servers. If a physical server serves more than one type of application, each application type should have a unique hash identifier.

The algorithm for computing the hash value for a service works as follows:

- By default, a global setting specifies the use of port number in a hash calculation.
- If you configure a hash identifier for a service, it is used, and the port number is not, regardless of the global setting.
- If you do not configure a hash identifier, but change the default value of the global setting so that it does not specify use of the port number, the hash value is based only on the IP address of the service.
- If you do not configure a hash identifier or change the default value of the global setting to use the port number, the hash value is based on the IP address and the port number of the service.

You can also specify hash identifiers when using the CLI to bind services to a service group. In the configuration utility, you can open a service group and add hash identifiers on the Members tab.

To change the use-port-number global setting by using the CLI

At the command prompt, type:

set lb parameter -usePortForHashLb (YES NO)

Example:

```
> set lb parameter -usePortForHashLb NO
 Done
>show lb parameter
Global LB parameters:
        Persistence Cookie HttpOnly Flag: DISABLED
        Use port for hash LB: NO
 Done
<!--NeedCopy-->
```

To change the use-port-number global setting by using the GUI

1. Navigate to Traffic Managment > Load Balancing > Configure Load Balancing parameters.
2. Select or clear Use Port for Hash Based LB Methods.

To create a new service and specify a hash identifier for a service by using the CLI

At the command prompt, type the following commands to set the hash ID and verify the setting:

add service < name > (< ip >	< serverName >) < serviceType > < port > -hashId < positive_integer >

```
show service <name>
<!--NeedCopy-->
```

Example:

```
> add service flbkng 10.101.10.1 http 80 -hashId 12345
 Done
>show service flbkng
        flbkng (10.101.10.1:80) - HTTP
        State: DOWN
        Last state change was at Thu Nov  4 10:14:52 2010
        Time since last state change: 0 days, 00:00:15.990
        Server Name: 10.101.10.1
        Server ID : 0   Monitor Threshold : 0

        Down state flush: ENABLED
        Hash Id: 12345

1)      Monitor Name: tcp-default
                State: DOWN     Weight: 1

 Done
<!--NeedCopy-->
```

To specify a hash identifier for an existing service by using the CLI

Type the set service command, the name of the service, and **-hashID** followed by the ID value.

To specify a hash identifier while adding a service group member

To specify a hash identifier for each member to be added to the group and verify the setting, at the command prompt, type the following commands (Be sure to specify a unique hashID for each member.):

```
bind servicegroup <serviceGroupName> <memberName> <port> -hashId <
   positive_integer>

show servicegroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
 bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
        SRV - HTTP
        State: ENABLED  Monitor Threshold : 0
        …

        1)          1.1.1.1:80   State: DOWN     Server Name: 1.1.1.1
              Server ID: 123  Weight: 1
        Hash Id: 32211

                Monitor Name: tcp-default       State: DOWN
        …

        2)          2.2.2.2:80   State: DOWN     Server Name: 2.2.2.2
              Server ID: 123  Weight: 1
        Hash Id: 12345

                Monitor Name: tcp-default       State: DOWN
        …
Done

<!--NeedCopy-->
```

To specify a hash identifier for a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. Create a new service, or open an existing service and specify the hash ID.

To specify a hash identifier for an already configured service group member by using the GUI

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a member and type a unique hash ID.

Configure the redirection mode

September 21, 2020

The redirection mode configures the method used by a virtual server to determine where to forward incoming traffic. The NetScaler appliance supports the following redirection modes:

- IP-Based forwarding (the default)
- MAC-Based forwarding

You can configure MAC-Based forwarding on networks that use direct server return (DSR) topology, link load balancing, or firewall load balancing. For more information on MAC-Based forwarding, see Configuring MAC-Based Forwarding.

To configure the redirection mode by using the CLI

At the command prompt, type:

```
set lb vserver <name> -m <RedirectionMode>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m MAC
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure the redirection mode by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server and select the redirection mode.

Configure per-VLAN wildcarded virtual servers

September 21, 2020

If you want to configure load balancing for traffic on a specific virtual local area network (VLAN), you can create a wildcarded virtual server with a listen policy that restricts it to processing traffic only on the specified VLAN.

To configure a wildcarded virtual server that listens to a specific VLAN by using the CLI

At the command prompt, type the following commands to configure a wildcarded virtual server that listens to a specific VLAN and verify the configuration:

```
add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <expression> [-listenpriority <positive_integer>]

show vserver
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)" -listenpriority 10

show vserver Vserver-LB-vlan1
<!--NeedCopy-->
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Create a new virtual server or open an existing virtual server.
3. Specify a listen policy priority and expression.

After you have created this virtual server, you bind it to one or more services as described in Setting Up Basic Load Balancing.

Assign weights to services

September 21, 2020

In a load balancing configuration, you assign weights to services to indicate the percentage of traffic that should be sent to each service. Services with higher weights can handle more requests; services with lower weights can handle fewer requests. Assigning weights to services allows the NetScaler appliance to determine how much traffic each load balanced server can handle, and therefore more effectively balance load.

Note: If you use a load balancing method that supports weighting of services (for example, the round robin method), you can assign a weight to the service.

The following table describes the load balancing methods that support weighting, and briefly describes the manner in which weighting affects how a service is selected for each one.

Load Balancing Methods	Service Selection with Weights
Round Robin	The virtual server prioritizes the queue of available services such that services with the highest weights come to the front of the queue more frequently than those with the lowest weights and receive proportionately more traffic. For a complete description, see The Round Robin Method.
Least Connection	The virtual server selects the service with the best combination of fewest active transactions and highest weight. For a complete description, see The Least Connection Method.

Load Balancing Methods	Service Selection with Weights
Least Response Time and Least Response Time Method using Monitors	The virtual server selects the service with the best combination of fewest active transactions and fastest average response time. For a complete description, see The Least Response Time Method.
Least Bandwidth	The virtual server selects the service with the best combination of least traffic and highest bandwidth. For a complete description, see The Least Bandwidth Method.
Least Packets	The virtual server selects the service with the best combination of fewest packets and highest weight. For a complete description, see The Least Packets Method.
Custom Load	The virtual server selects the service with the best combination of lowest load and highest weight. For a complete description, see The Custom Load Method.
Hashing methods and Token method	Weighting is not supported by these load balancing methods.

To configure a virtual server to assign weights to services by using the CLI

At the command prompt, type:

```
set lb vserver <name> -weight <Value> <ServiceName>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
<!--NeedCopy-->
```

To configure a virtual server to assign weights to services by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open the virtual server, and then click in the **Services** section.
3. In the weight column for the service, assign a weight to the service.

Configure the MySQL and Microsoft SQL server version setting

September 21, 2020

You can specify the version of Microsoft® SQL Server® and the MySQL server for a load balancing virtual server that is of type MSSQL and MySQL respectively. The version setting is recommended if you expect some clients to not be running the same version as your MySQL or Microsoft SQL Server product. The version setting provides compatibility between the client-side and server-side connections by ensuring that all communication conforms to the server's version.

To set the Microsoft SQL server version parameter by using the CLI

At the command prompt, type the following commands to set the Microsoft SQL Server version parameter for a load balancing virtual server and verify the configuration:

```
set lb vserver <name> -mssqlServerVersion <mssqlServerVersion>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver myMSSQLvip -mssqlServerVersion 2008R2
 Done
> show lb vserver myMSSQLvip
    myMSSQLvip (190.0.2.12:1433) - MSSQL    Type: ADDRESS
 . . .
 . . .
    MSsql Server Version: 2008R2
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

To set the MySQL server version parameter by using the CLI

At the command prompt, type the following commands to set the MySQL Server version parameter for a load balancing virtual server and verify the configuration:

```
set lb vserver <name> -mysqlServerVersion <string>
```

```

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
> set lb vserver mysqlsvr -mysqlserverversion 5.5.30
 Done
> sh lb vserver mysqlsvr
     mysqlsvr (2.22.2.222:3306) - MYSQL      Type: ADDRESS
 . . .
 . . .
     Mysql Server Version: 5.5.30
    . . .
    . . .
 Done
>
<!--NeedCopy-->
```

To set the MySQL or Microsoft SQL server version parameter by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open a virtual server of type MySQL or MSSQL, and set the server version.

Configure diameter load balancing

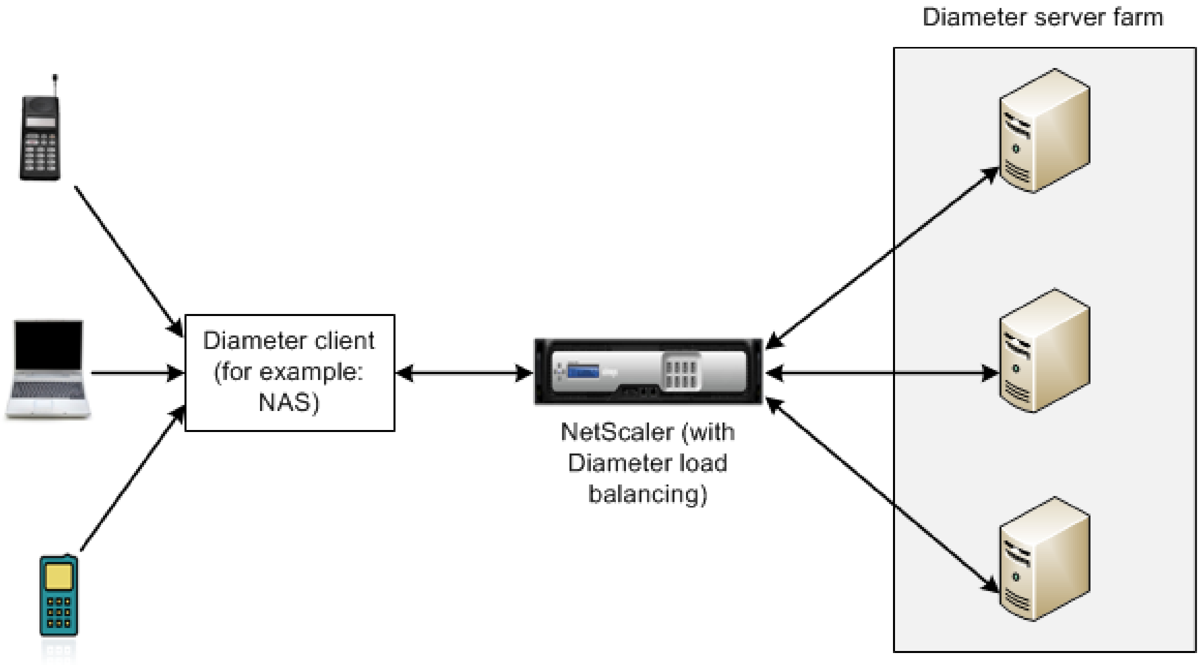
September 21, 2020

The Diameter protocol is a next generation Authentication, Authorization, and Accounting (AAA) signaling protocol used mainly on mobile devices such as laptops and mobile phones. It is a peer-to-peer protocol, as opposed to the traditional client-server model used by most other protocols. However, in most Diameter deployments, the clients originates the request and the server responds to the request.

When Diameter messages are exchanged, the Diameter server usually does much more processing than does the Diameter client. With the increase in control plane signaling volume, the Diameter server becomes a bottleneck. Therefore, Diameter messages must be load balanced to multiple servers. A virtual server performing load balancing of Diameter messages provides the following benefits:

- Lighter load on Diameter servers, which translates to faster response time to end users.
- Server health monitoring and better failover capabilities.
- Better scalability in terms of server addition without changing client configuration.
- High availability.
- SSL-Diameter offloading.

The following figure shows a Diameter system in a NetScaler deployment:


Diameter server farm
Diameter client (for example: NAS)
NetScaler (with Diameter load balancing)

A Diameter system has the following components:

- **Diameter client.** Supports Diameter client applications in addition to the base protocol. Diameter clients are often implemented in devices situated at the edge of a network and provide access control services for that network. Typical examples of Diameter clients are a Network Access Server (NAS) and the Mobile IP Foreign Agent (FA).
- **Diameter agent.** Provides relay, proxy, redirect, or translation services. The NetScaler appliance (configured with a Diameter load balancing virtual server) plays the role of a Diameter agent.
- **Diameter server.** Handles the authentication, authorization, and accounting requests for a particular realm. A Diameter server must support Diameter server applications in addition to the base protocol.

In a typical Diameter topology, when an end-user device (such as a mobile phone) needs a service, it sends a request to a Diameter client. Each Diameter client establishes a single connection (TCP connection—SCTP is not yet supported) with a Diameter server as specified by the Diameter base-protocol RFC 6733. The connection is long-lived and all messages between the two Diameter nodes (client and server) are exchanged over this connection. The NetScaler uses message based load balancing.

Example:

A mobile service provider uses Diameter for its billing system. When a subscriber uses a prepaid number, the Diameter client repeatedly sends requests to the server to check the available balance. The Diameter protocol establishes a connection between the client and the server, and all requests are exchanged over that connection. Connection based load balancing would be pointless, because there is only one connection. However, with the large number of messages on the connection, message based load balancing expedites the process of billing the prepaid mobile subscriber.

How diameter load balancing works

A Disconnect Peer Request (DPR) indicates the peer's intention of closing the connection, with the reason for closing the connection. The peer replies with a DPA (TCP always provides successful DPA).

- When the appliance receives a DPR from the client, it broadcasts the DPR to all servers and immediately replies with a DPA to the client. The servers reply with DPAs, but the appliance ignores them. The client sends a FIN, which the appliance broadcasts to all servers.
- When the appliance receives a DPR from the server, it replies with a DPA to that server alone, and does not remove the server from the reuse pool. When the server sends a FIN, the appliance replies with FIN/ACK and removes connections from the reuse pool.
- If the appliance receives a FIN from the client, it sends the client a FIN/ACK, broadcasts the FIN, and immediately removes the server connection from the reuse pool.
- If the appliance receives a FIN from the server, it sends a FIN/ACK and removes it from reuse pool. Any new message for this server is sent on a new connection.

Load balancing diameter traffic

When a client sends a request to the NetScaler appliance, the appliance parses the request and contextually load balances it to a Diameter server on the basis of a persist AVP. The appliance has advertised the client identity to the server, so it does not add route entries, because the server is expecting messages directly from client.

Server initiated requests are not as frequent as client requests. Server initiated requests are similar to client initiated requests, except:

- Since messages are received from multiple servers, the appliance maintains the transaction state by adding a unique Hop by Hop (HbyH) number to each forwarded request message. When the message response arrives (with same HbyH number), the appliance translates this HbyH number to the HbyH number that was received on the server when the request arrived.
- The NetScaler appliance adds a route entry by putting its identity, because the client sees the appliance as a relay agent.

Note: If a Diameter message spans more than one packet, the appliance accumulates the packets in an incomplete header queue and forwards them to the server when the full message is accumulated. Similarly, if a single packet contains more than one Diameter message, the appliance splits the packet and forwards the messages to servers as determined by the load balancing virtual server.

Disconnect a session

A Disconnect Peer Request (DPR) indicates the peer's intention of closing the connection, with the reason for closing the connection. The peer replies with a DPA (TCP always provides successful DPA).

- When the NetScaler appliance receives a DPR from the client, it broadcasts the DPR to all servers and immediately replies with a DPA to the client. The servers reply with DPAs, but the appliance ignores them. The client sends a FIN, which the appliance broadcasts to all servers.
- When the appliance receives a DPR from the server, it replies with a DPA to that server alone, and does not remove the server from the reuse pool. When the server sends a FIN, the appliance replies with FIN/ACK and removes connections from the reuse pool.
- If the appliance receives a FIN from the client, it sends the client a FIN/ACK, broadcasts the FIN, and immediately removes the server connection from the reuse pool.
- If the appliance receives a FIN from the server, it sends a FIN/ACK and removes it from reuse pool. Any new message for this server is sent on a new connection.

Configure load balancing for diameter traffic

To configure the NetScaler appliance to load balance diameter traffic, you must first set the Diameter parameters on the appliance, then add the diameter monitor, add the diameter services, bind the services to the monitor, add the diameter load balancing virtual server, and bind the services to the virtual server.

To configure load balancing for diameter traffic by using the command line interface

Configure the diameter parameters.

```
set ns diameter -identity <string> -realm <string> -serverClosePropagation <YES|NO>
<!--NeedCopy-->
```

Example:

```
set ns diameter -identity mydomain.org -realm org -serverClosePropagation YES
<!--NeedCopy-->
```

Add a Diameter monitor.

```
add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm <string>
<!--NeedCopy-->
```

Example:

```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -originRealm org
<!--NeedCopy-->
```

Create the Diameter services.

```
add service <name> <IP> DIAMETER <port>
<!--NeedCopy-->
```

Example:

```
add service diameter_svc0 10.102.82.86 DIAMETER 3868

add service diameter_svc1 10.102.82.87 DIAMETER 3868

add service diameter_svc2 10.102.82.88 DIAMETER 3868

add service diameter_svc3 10.102.82.89 DIAMETER 3868
<!--NeedCopy-->
```

Bind the Diameter services to the Diameter monitor.

```
bind service <name>@ monitorName <monitorName>
<!--NeedCopy-->
```

Example:

```
bind service diameter_svc0 -monitorName diameter_mon

bind service diameter_svc1 -monitorName diameter_mon

bind service diameter_svc2 -monitorName diameter_mon

bind service diameter_svc3 -monitorName diameter_mon
<!--NeedCopy-->
```

Add a Diameter load balancing virtual server with Diameter persistence.

```
add lb vserver <name> DIAMETER <IPAddress> <port> -
    persistenceType DIAMETER -persistAVPno <positive_integer>
<!--NeedCopy-->
```

Example:

```
add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -
    persistenceType DIAMETER -persistAVPno 263
<!--NeedCopy-->
```

Bind the Diameter services to the Diameter load balancing virtual server.

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver diameter_vs diameter_svc0

bind lb vserver diameter_vs diameter_svc1

bind lb vserver diameter_vs diameter_svc2

bind lb vserver diameter_vs diameter_svc3
<!--NeedCopy-->
```

Save the configuration.

```
save ns config
<!--NeedCopy-->
```

Note: You can also configure load balancing of Diameter traffic over SSL by using the **SSL_DIAMETER** service type.

To configure load balancing for Diameter traffic by using the configuration utility

1. Navigate to **System** > **Settings** > **Change Diameter Parameters** and set the diameter parameters.
2. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and create a load balancing virtual server of type Diameter.
3. Create a service of type Diameter.
4. Create a monitor of type Diameter. In Special parameters, set the origin host and origin realm.
5. Bind the monitor to the service, and bind the service to the Diameter virtual server.

6. In Advanced Settings, click **Persistence**, specify Diameter and enter a persistence AVP number.
7. Click **Save**, and click **Done**.

Configure FIX load balancing

September 21, 2020

Financial Information eXchange (FIX) protocol is an open message standard used in financial industry for electronic exchange of information related to securities transaction between trading partners. FIX/SSL_FIX protocol is used extensively by buy-side and sell-side firms, trading platforms, and regulators for communicating trade information.

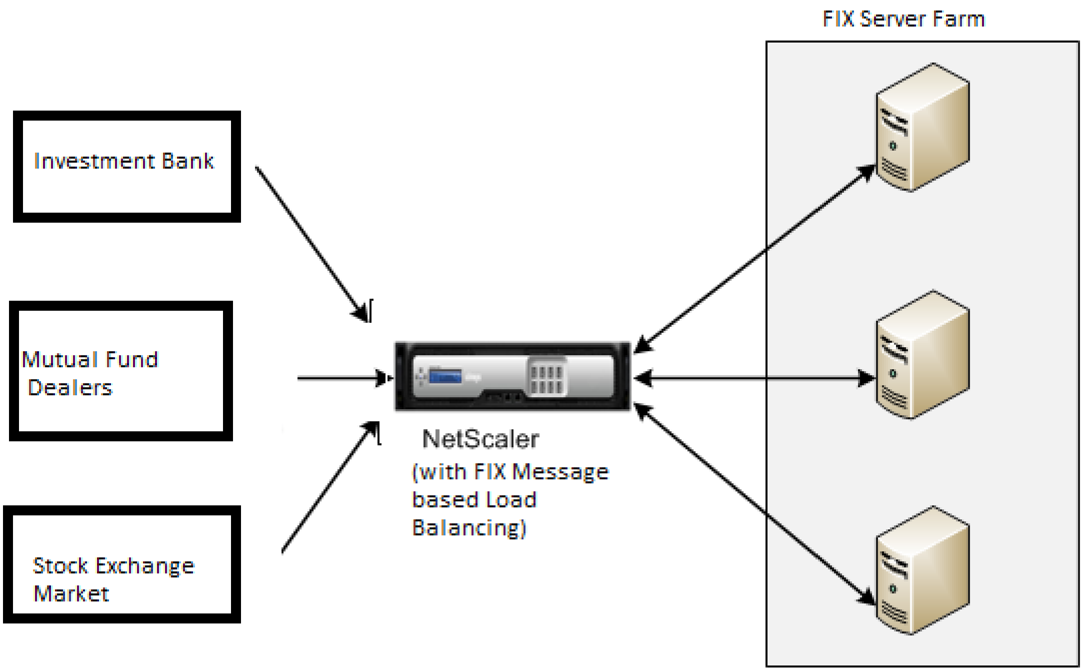
This feature enables you to configure a FIX or SSL_FIX load balancing virtual server to distribute incoming FIX messages and provide security in FIX messaging. NetScaler supports FIX message based load balancing (MBLB) for FIX 4.1, FIX 4.2, FIX 4.3 and FIX 4.4 versions.

FIX MBLB on a NetScaler appliance provides the following benefits:

1. Efficient management of FIX or SSL_FIX servers with superior HA and health monitoring.
2. SYN protection to all FIX or SSL_FIX servers.
3. FIX session persistence.

How FIX load balancing works

A FIX MBLB setup includes a FIX load-balancing virtual server and multiple load-balanced FIX servers. The FIX virtual server receives incoming client traffic, parses the incoming traffic into FIX messages, selects a FIX server for each FIX message and forwards the message to the selected FIX server. The following conceptual drawing illustrates a typical FIX load balancing set up.


FIX Server Farm
Investment Bank
Mutual Fund Dealers
Stock Exchange Market
NetScaler
(with FIX Message based Load Balancing)

In a basic FIX MBLB setup, the FIX virtual server distributes FIX messages coming from clients to the load-balanced FIX servers using the round robin load-balancing method. With persistence of type FIXSESSION enabled, the FIX virtual server selects the same server for different FIX messages belonging to the same FIX session. The FIX session is determined based on the values of FIX fields SenderCompID (tag 49) and TargetCompID (tag 56).

Configure and monitor load balancing for FIX traffic

Following are the configurations that you must do to load balance FIX message traffic:

1. Configuring FIX load balancing virtual server
2. Configuring SSL_FIX load balancing virtual server
3. Configuring FIX load balancing service
4. Configuring SSL_FIX load balancing service
5. Configuring FIXSESSION persistence
6. Setting persistence timeout
7. Displaying FIX/SSL_FIX stats
8. Monitoring FIX/SSL_FIX persistent sessions

To configure a FIX load balancing server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> FIX <IP> <PORT>
```

```
<!--NeedCopy-->
```

Example

```
add lb vserver vs1 FIX 10.102.82.86 3868
<!--NeedCopy-->
```

To configure a SSL_FIX load balancing virtual server by using the command line interface

At the command prompt, type:

```
add lb vserver <name> SSL_FIX <IP> <PORT>
<!--NeedCopy-->
```

Example

```
add lb vserver vs1 SSL_FIX 10.102.82.86 3868
<!--NeedCopy-->
```

To configure a FIX service by using the command line interface

At the command prompt, type:

```
add service <name> <ip-addr> FIX <port>
<!--NeedCopy-->
```

Example

```
add service_svc1 10.102.82.86 FIX 3868
<!--NeedCopy-->
```

To configure a SSL_FIX service by using the command line interface

At the command prompt, type:

```
add service <name> <ip-addr> SSL_FIX <port>
<!--NeedCopy-->
```

Example

```
add service svc1 10.102.82.86 SSL_FIX 3868
<!--NeedCopy-->
```

To configure FIXSESSION persistence by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -persistenceType FIXSESSION
<!--NeedCopy-->
```

Example

```
set lb vserver vs1 -persistenceType FIXSESSION
<!--NeedCopy-->
```

To set persistence timeout by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -timeout <value>
<!--NeedCopy-->
```

Example

```
set lb vserver vs1 –timeout 2
<!--NeedCopy-->
```

To display FIX stats by using the command line interface

At the command prompt, type:

```
stat lb vserver <name>
<!--NeedCopy-->
```

Example

```
stat lb vserver_svc1
<!--NeedCopy-->
```

To bind FIX service to FIX virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name> <service name>
<!--NeedCopy-->
```

Example

```
bind lb vserver vs1 svc1
<!--NeedCopy-->
```

To display FIX persistent sessions by using the command line interface

At the command prompt, type:

```
show lb persistentSessions <name>
<!--NeedCopy-->
```

Example

```
show lb persistentSessions vs1
<!--NeedCopy-->
```

> **Note**
>
> Note: You can now configure the load balancing of FIX traffic over SSL by using the SSL_FIX service type. This service provides secured communication for FIX messages.

To configure FIX load balancing virtual server by using the GUI

1. Navigate to the **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and click **Add** to create a FIX Load Balancing virtual server.
2. On the **Load Balancing Virtual Server** page, set the server parameters:
   a) Virtual Server Name
   b) Protocol type as "FIX"
   c) Server IP Address Type
   d) Server IP Addres
   e) Server Port Number
3. Click **OK** and **Continue** to set additional parameters.
4. In the **Services** section, select or add a new FIX load balancing virtual service, and bind it to the FIX server.
5. In the **Persistence** section, set the following parameters:
   a) Persistence type as 'FIXSESSION'
   b) Time-out interval
6. Click **OK** and then **Done**.

To edit a FIX load balancing virtual server by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server and click **Edit**.

To delete a FIX load balancing virtual server by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page, select a FIX server and click **Delete**.

To configure FIX Load Balancing Virtual Service by using the GUI

1. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page and click **Add** to create a FIX Load Balancing virtual service.
2. On the **Services** page, set the following parameters. You can click the 'More' arrow to set additional parameters such as Traffic Domain, Hash ID, Server ID, Cache Type, and Number of Active Connections.
   a) Service Name – FIX Virtual Service Name
   b) Choose Virtual Server type as (New or Existing)
   c) Protocol – Protocol Type as 'FIX'
   d) Server – Virtual Server IP address
   e) Port – Server Port Number
3. Click **OK** and **Continue** to set other parameters such as Monitors, Threshold & Timeout, Profiles, and Policies.
4. Click **OK** and then **Done**.

To edit a FIX load balancing virtual service by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a **FIX service** and click **Edit**.

To delete a FIX load balancing virtual service by using the GUI

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** page, select a FIX service and click **Delete**.

To display FIX load balancing server statistics

Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** page and then click **Statistics** to display the FIX server statistics.

To display Persistent sessions for a FIX server by using the GUI

Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Virtual Server Persistent Sessions**.

To clear Persistent sessions for a FIX server by using the GUI

1. Navigate to **Configuration** > **Traffic Management** page and, under **Monitor Sessions** click **Clear Persistent Sessions**.
2. On the **Clear Persistent Sessions** page, set the following parameters:
    a) Virtual Server – Choose a FIX virtual server
    b) Persistence Parameter – Choose a FIX persistence parameter
3. Click **OK**.

Protect a load balancing configuration against failure

September 21, 2020

When a load balancing virtual server fails, or when the virtual server is unable to handle excessive traffic, the load balancing setup can fail. You can protect your load balancing setup against failure by configuring the NetScaler appliance to redirect excess traffic to an alternate URL, configuring a backup load balancing virtual server, and configuring stateful connection failover.

Redirect client requests to an alternate URL

September 21, 2020

In the event that a load balancing virtual server of type HTTP or type HTTPS goes DOWN or is disabled, you can redirect requests to an alternate URL by using an HTTP 302 redirect. The alternate URL can provide information about the status of the server.

You can redirect to a page on the local server or a remote server. You can redirect to a relative URL or an absolute URL. If you configure a redirect to a relative URL consisting of a domain name with no path, the NetScaler appliance appends the path of the incoming URL to the domain. If you use an absolute URL, the HTTP redirect is sent to that URL with no modification.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when both the primary and backup virtual servers are DOWN.

To configure a virtual server to redirect the client request to a URL by using the CLI

At the command prompt, type:

```
set lb vserver <vServerName> -redirectURL <URLValue>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -redirectURL http://www.newdomain.com/
   mysite/maintenance
<!--NeedCopy-->
```

To configure a virtual server to redirect the client request to a URL by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and specify a redirect URL.

Configure a backup load balancing virtual server

March 9, 2021

You can configure the NetScaler appliance to direct requests to a backup virtual server in the event that the primary load balancing virtual server is DOWN or unavailable. The backup virtual server is a proxy and is transparent to the client. The appliance can also send a notification message to the client regarding the site outage.

> Note:
>
> The backup virtual server continues to handle the existing connections, even after the primary virtual server is deleted or disabled.

You can configure a backup load balancing virtual server when you create it, or you can change the optional parameters of an existing virtual server. You can also configure a backup virtual server for an existing backup virtual server, thus creating cascading backup virtual servers. The maximum depth of cascading backup virtual servers is 10.

If you have multiple virtual servers that connect to two servers, you have a choice for what happens if the primary virtual server goes DOWN and then comes back up. The default behavior is for the primary virtual server to resume its role as primary. However, you may want to configure the backup virtual server to remain in control in the event that it takes over. For example, you may want to sync updates on the backup virtual server to the primary virtual server and then manually force the original primary server to resume its role. In this case, you can designate the backup virtual server to remain in control in the event that the primary virtual server goes DOWN and then comes back up.

You can configure a redirect URL on the primary load balancing virtual server as a fallback for when both the primary and the backup virtual servers are DOWN or have reached their threshold for handling requests. When services bound to virtual servers are OUT OF SERVICE, the appliance uses the redirect URL.

Note: If a load balancing virtual server is configured with both a backup virtual server and a redirect URL, the backup virtual server takes precedence over the redirect URL. A redirect is used only when the primary and backup virtual servers are down.

To set a backup virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-
    disablePrimaryOnDown]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -
    disablePrimaryOnDown
<!--NeedCopy-->
```

To set a backup virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and select a backup virtual server.
3. If you want the backup virtual server to remain in control until you manually enable the primary virtual server even if the primary virtual server comes back up, select **Disable Primary When Down**.

Configure spillover

September 21, 2020

A spillover configuration on the appliance consists of a primary virtual server that is configured with a spillover method, a spillover threshold, and a backup virtual server. Backup virtual servers can also be configured for spillover, creating a chain of backup virtual servers.

The spillover method specifies the operational condition on which you want to base your spillover configuration (for example, the number of established connections, bandwidth, or combined health of the server farm). When a new connection arrives, the appliance verifies that the primary virtual server is up and compares the operational condition with the configured spillover threshold. If the threshold is reached, the spillover feature diverts new connections to the first available virtual server in the backup chain. The backup virtual server manages the connections it receives until the load on the primary falls below the threshold.

If you configure spillover persistence, the backup virtual server continues to process the connections it received, even after the load on the primary falls below the threshold. If you configure spillover persistence and a spillover persistence timeout, the backup virtual server processes connections for only the specified period after the load on the primary falls below the threshold.

Note: Usually, spillover is triggered if the value associated with the spillover method exceeds the threshold (for example, number of connections). However with the server-health spillover method, spillover is triggered if the health of the server farm falls below the threshold.

You can configure spillover in one of the following ways:

- Specify a predefined spillover method. Four predefined methods are available, and they fulfill common spillover requirements.
- Configure policy based spillover. In policy based spillover, you use a NetScaler rule to specify the conditions that should be met for spillover to occur. NetScaler rules give you the flexibility to configure spillover for various operational conditions.

Use policy based spillover if a predefined method does not satisfy your requirements. If you configure both for a primary virtual server, the policy based spillover configuration takes precedence over the predefined method.

First, you create the primary virtual server and the virtual servers that you need for the backup chain. You set up the backup chain by specifying one virtual server as the backup for the primary (that is, you create a secondary virtual server), a virtual server as the backup for the secondary (that is, you create a tertiary virtual server), and so on. Then, you configure spillover by either specifying a predefined spillover method or creating and binding spillover policies.

For instructions for assigning a virtual server as the backup for another virtual server, see Configuring a Backup Load Balancing Virtual Server.

Configure a predefined spillover method

Predefined spillover methods fulfill some of the more common spillover requirements. To use one of the predefined spillover methods, you configure spillover parameters on the primary virtual server. To create a chain of backup virtual servers, you also configure spillover parameters on backup virtual servers.

If the backup virtual servers reach their own threshold values, and the service type is TCP, the NetScaler appliance sends clients a TCP reset. For service types HTTP, SSL, and RTSP, it diverts new requests to the redirect URL configured for the primary virtual server. A redirect URL can be specified for only HTTP, SSL, and RTSP virtual servers. If a redirect URL is not configured, the NetScaler appliance sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).

Note: With RTSP virtual servers, the NetScaler appliance uses only data connections for spillover. If the backup RTSP virtual server is not available, the requests are redirected to an RTSP URL and an RTSP redirect message is sent to the client.

To configure a predefined spillover method for a virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -soMethod <spillOverType> -soThreshold <
    positiveInteger> -soPersistence ENABLED -soPersistenceTimeout <
    positiveInteger>
<!--NeedCopy-->
```

Example

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -
    soPersistence enabled -soPersistenceTimeout 2
<!--NeedCopy-->
```

To configure a predefined spillover method for a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and set the spillover parameters.

Configure policy based spillover

Spillover policies, which are based on rules (expressions), enable you to configure the appliance for a wider range of spillover scenarios. For example, you can configure spillover based on the virtual server's response time, or based on number of connections in the virtual server's surge queue.

To configure policy based spillover, first create a spillover action. You then select the expression that you want to use in the spillover policy, configure the policy, and associate the action with it. Finally, you bind the spillover policy to a load balancing, content switching, or global server load balancing virtual server. You can bind multiple spillover policies to a virtual server, with priority numbers. The appliance evaluates the spillover policies in ascending order of priority numbers and performs the action associated with the last policy to evaluate to TRUE.

A virtual server can also have a backup action. The backup action is performed if the virtual server does not have one or more backup virtual servers, or if all of the backup virtual servers are DOWN, disabled, or have reached their own spillover limits.

When a spillover policy results in an UNDEF condition (an exception thrown when the result of policy evaluation is undefined), an UNDEF action is performed. The UNDEF action is always ACCEPT. You cannot specify an UNDEF action of your choice.

Configuring a Spillover Action

A spillover action is performed when the spillover policy with which it is associated evaluates to TRUE. Currently, SPILLOVER is the only supported spillover action.

To configure policy based spillover by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

```
add spillover action <name> -action SPILLOVER

show spillover action <name>
<!--NeedCopy-->
```

Example

```
add spillover action mySoAction -action SPILLOVER
Done
<!--NeedCopy-->
```

```
show spillover action mySoAction
1) Name: mySoAction Action: SPILLOVER
Done
<!--NeedCopy-->
```

Selecting an Expression for the Spillover Policy

In the policy expression, you can use any virtual-server based expression that returns a Boolean value. For example, you can use one of the following expressions:

```
SYS.VSERVER("vserver").RESPTIME.GT(<int>)
SYS.VSERVER("vserver").STATE.EQ( “<string>” ), and
SYS.VSERVER("vserver").THROUGHPUT.LT    (<int>)
<!--NeedCopy-->
```

In addition to the existing functions such as RESPTIME, STATE, and THROUGHPUT, you can use the following virtual server based functions that have been introduced with this feature:

Averagesurgecount

Returns the average number of requests in the surge queues of active services. Returns 0 (zero) if there are no active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

Activeservices

Returns the number of active services. Raises an UNDEF condition if used with a content switching or global server load balancing virtual server.

Activetransactions

Returns the value of the virtual-server-level counter for current active transactions.

is_dynamic_limit_reached

Returns a Boolean TRUE if the number of connections that the virtual server manages is equal to the dynamically calculated threshold. The dynamic threshold is the sum of the maximum client (Max Clients) settings of the bound services that are UP.

You can use a policy expression to implement any of the predefined spillover methods. The following table maps the predefined spillover methods to the expressions you can use to implement them:

Table 1. Converting predefined spillover methods to policy expressions

Predefined spillover method	Corresponding expression
CONNECTION	SYS.VSERVER(“`<vserver-name>`”).CONNECTIONS, used with the GT(int) arithmetic function.
BANDWIDTH	SYS.VSERVER(“`<vserver-name>`”).THROUGHPUT, used with the GT(int) arithmetic function.
HEALTH	SYS.VSERVER(“`<vserver-name>`”).HEALTH, used with the LT(int) arithmetic function.
DYNAMICCONNECTION	SYS.VSERVER(“`<vserver-name>`”).IS_DYNAMIC_LIMIT_REACHED **Note**: If you implement policy based spillover by using the IS_DYNAMIC_LIMIT_REACHED function, you must also configure the predefined DYNAMICCONNECTION method for the virtual server, so that statistics required for spillover to work are collected.

Configuring a Spillover Policy

A spillover policy uses a Boolean expression as a rule to specify the conditions that must be met for spillover to occur.

To configure a spillover policy by using the command line interface

At the command prompt, type the following commands to configure a spillover policy and verify the configuration:

```
add spillover policy <name> -rule <expression> -action <string> [-comment <string>]

show spillover policy <name>
<!--NeedCopy-->
```

Example

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT(50) -action mySoAction -comment "Triggers spillover when the vserver's response time is greater than 50 ms."
```

```
Done
<!--NeedCopy-->
```

```
> show spillover policy mySoPolicy

Name: mySoPolicy Rule: "SYS.VSERVER("v1").RESPTIME.GT(50)" Action:
    mySoAction Hits: 0 ActivePolicy: 0
Comment: "Triggers spillover when the vserver's response time is
    greater than 50 ms."
Done
<!--NeedCopy-->
```

Binding a Spillover Policy to a Virtual Server

You can bind a spillover policy to load balancing, content switching, or global server load balancing virtual servers). You can bind multiple policies to a virtual server, with Goto expressions controlling the flow of evaluation.

To bind a spillover policy to a virtual server by using the command line interface

At the command prompt, type the following commands to bind a spillover policy to a load balancing, content switching, or global server load balancing virtual server and verify the configuration:

```
bind (lb | cs | gslb) vserver <name> -policyName <string> -priority <
    positive_integer> [-gotoPriorityExpression <expression>]

show (lb | cs | gslb) vserver <name>
<!--NeedCopy-->
```

Example

> bind lb vserver vserver1 -policyName mySoPolicy -priority 5
> Done
> show lb vserver vserver1
> vserver1 (2.2.2.12:80) - HTTP Type: ADDRESS
> ...

1) Spillover Policy Name: mySoPolicy Priority: 5
GotoPriority Expression: END
Flowtype: REQUEST
Done

Configuring a Backup Action for a Spillover Event

A backup action specifies what to do when the spillover threshold is reached but one or more backup virtual serves are either not configured or are down, disabled, or have reached their own thresholds.

Note: For the predefined spillover methods that are configured directly on the virtual server (as values of the Spillover Method parameter), the backup action is not configurable. By default, the appliance sends clients a TCP reset (if the virtual server is of type TCP) or an HTTP 503 response (if the virtual server is of type HTTP or SSL).

The backup action is configured on the virtual server. You can configure the virtual server to accept requests (after the threshold specified by the policy is reached), redirect clients to a URL, or simply drop requests even before establishing TCP or SSL connections until the number of requests falls below the threshold. Therefore, lesser memory resources are utilized as the connections are reset even before allocating any data structures.

To configure a backup action for spillover by using the CLI

At the command prompt, type the following commands to configure a backup action and verify the configuration:

```
set lb vserver <name> -soBackupAction <soBackupAction>

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
set lb vserver vs1 -soBackupAction REDIRECT -redirectURL http://www.mysite.com/maintenance
Done
show lb vserver vs1
vs1 (10.102.29.76:80) - HTTP Type: ADDRESS
State: UP
...
Redirect URL: http://www.mysite.com/maintenance
...
Done
```

To configure a backup action for spillover by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Protection**, and then specify a spillover backup action.

Connection failover

May 24, 2021

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. In a NetScaler High Availability (HA) setup, *connection failover* (or *connection mirroring-CM*) refers to keeping active an established TCP or UDP connection when a failover occurs. The new primary NetScaler appliance has information about the connections established before the failover and continues to serve those connections. After failover, the client remains connected to the same physical server. The new primary appliance synchronizes the information with the new secondary appliance. If the L2Conn parameter is set, Layer 2 connection parameters are also synchronized with the secondary.

> Note:
>
> Consider a HA setup, where a client establishes a session with the primary node, which in turn establishes a session with the back end server. When a failover is triggered in this state, the packets received on a new primary from the existing client and server nodes are treated as stale packets, and the client and server connections are reset. Whereas if stateless connection failover is enabled (USIP is ON), after the failover, the connections are not reset when you receive packets from client or server nodes. Instead, the client and server connections are created dynamically.

You can set up connection failover in either stateless or stateful mode. In the stateless connection failover mode, the HA nodes do not exchange any information about the connections that are failed over. This method has no runtime overhead.

In the stateful connection failover mode, the primary appliance synchronizes the data of the failed-over connections with the new secondary appliance.

Connection failover is helpful if your deployment has long lasting connections. For example, if you are downloading a large file over FTP and a failover occurs during the download, the connection breaks and the download is aborted. However, if you configure connection failover in stateful mode, the download continues even after the failover.

How connection failover works on NetScaler appliances

In stateless connection failover, the new primary appliance tries to re-create the packet flow according to the information contained in the packets it receives.

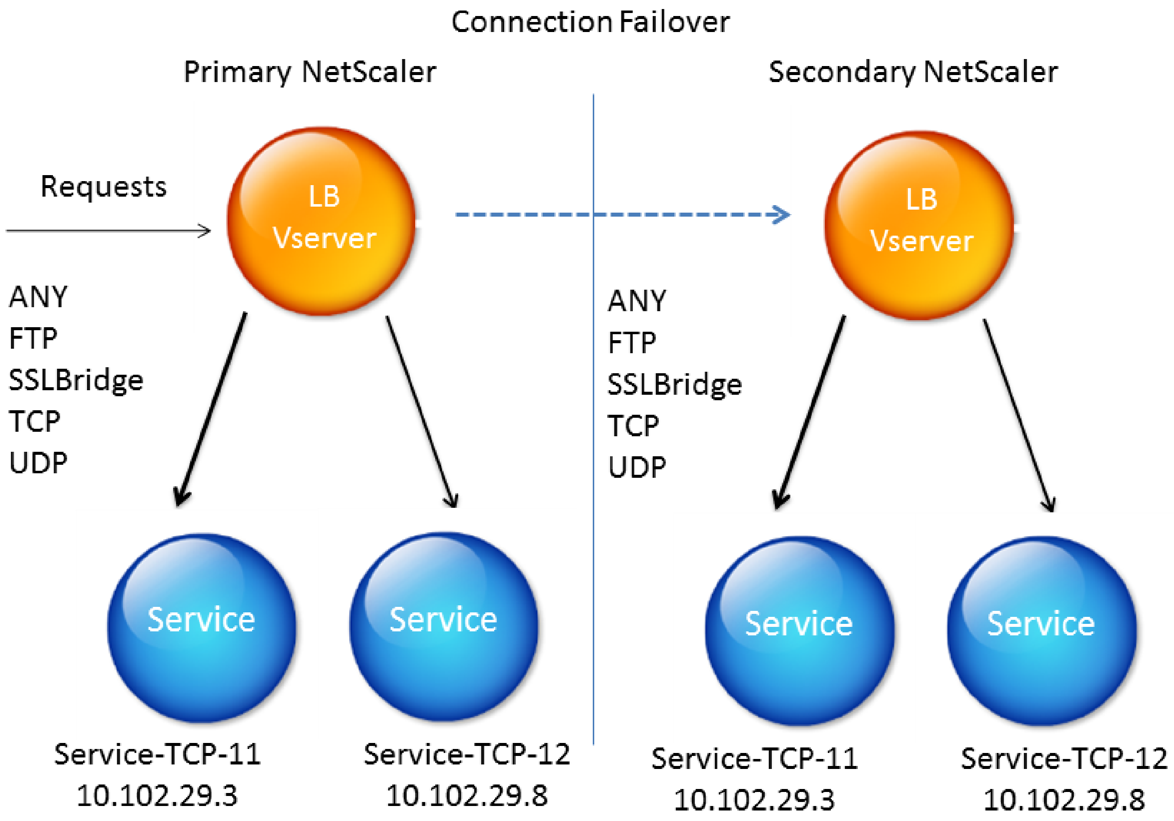
In stateful failover, to maintain current information about the mirrored connections, the primary appliance sends messages to the secondary appliance. The secondary appliance maintains the data related to the packets but uses it only in the event of a failover. If a failover occurs, the new primary (old secondary) appliance starts using the stored data about the mirrored connections and accepting traffic. During the transition period, the client and server may experience a brief disruption and retransmissions.

Note:

Verify that the primary appliance is able to authorize itself on the secondary appliance. To verify correct configuration of the passwords, use the show rpcnode command from command line or use the RPC option of the Network menu in GUI.

A basic HA configuration with connection failover contains the entities shown in the following figure.

Figure 1. Connection Failover Entity Diagram


Connection Failover
Primary NetScaler
Secondary NetScaler
Requests
LB Vserver
LB Vserver
ANY
FTP
SSLBridge
TCP
UDP
ANY
FTP
SSLBridge
TCP
UDP
Service
Service
Service
Service
Service-TCP-11
10.102.29.3
Service-TCP-12
10.102.29.8
Service-TCP-11
10.102.29.3
Service-TCP-12
10.102.29.8

> **Note**
>
> Connection failover is not supported after either of the following events:
>
> ```
> - An upgrade to a later release.
>  - An upgrade to a later build within the same release, if the new build uses a different HA version.
> ```

Supported setup

Connection failover can be configured only on load balancing virtual servers. It cannot be configured on content switching virtual servers. If you enable connection failover on load balancing virtual servers that are attached to a content switching virtual server, connection failover will not work because the load balancing virtual servers do not initially accept the traffic.

The following table describes the setup supported for connection failover.

Table 1. Connection Failover - Supported Setup

Setting	Stateless	Stateful
Service type	ANY.	ANY, UDP, TCP, FTP, SSL_BRIDGE.
Load balancing methods	All methods supported for the service type ANY. However, if Source IP persistence is not set, the SRCIPSRCPORTHASH method must be used.	All methods applicable to the supported service types.
Persistence types	SOURCEIP persistence.	All types applicable to the supported service types are supported.
USIP	Must be ON.	No restriction. It can be ON or OFF.
Service bindings	Service can be bound to only one virtual server.	Service can be bound to one or more virtual servers.
Internet Protocol (IP) versions	IPv4 and IPv6	IPV4 and IPv6
Redundancy support	Clustering and high availability	High availability

Features affected by connection failover

The following table lists the features affected if connection failover is configured.

Table 2. How Connection Failover Affects NetScaler Features

Feature	Impact of Connection Failover
SYN protection	For any connection, if a failover occurs after the appliance issues SYN-ACK but before it receives the final ACK, the connection is not supported by connection failover. The client must reissue the request to establish the connection.
Surge protection	If the failover occurs before a connection with the server is established, the new primary appliance tries to establish the connection with the server. It also retransmits all the packets held in the course of surge protection.
Access down	If enabled, the access-down functionality takes precedence over connection failover.
Application firewall	The Application firewall feature is not supported.
INC	Independent network configuration is not supported in the high availability (HA) mode.
TCP buffering	TCP buffering is not compatible with connection mirroring.
Close on response	After failover, the NATPCBs may not be closed on response.

To configure connection failover by using GUI

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**. Open the virtual server, and in **Advanced Settings**, click **Protection**, and select **Connection Failover** as **Stateful**.

To configure connection failover by using CLI

At the command prompt:

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
```

Example:

> set lb vserver Vserver-LB-1 -connFailover stateful
>
> Done

When connection failover is disabled on a virtual server, the resources allocated to the virtual server are freed.

To disable connection failover by using CLI

At the command prompt:

```
set lb vserver <vServerName> -connFailover <Value>
show lb vserver <vServerName>
```

Example:

> set lb vserver Vserver-LB-1 -connFailover disable
>
> Done

To disable connection failover by using GUI

Navigate to **Traffic Management**> **Load Balancing**> **Virtual Servers**. Open the virtual server, in **Protection**, select **Connection Failover** as Disabled.

Flush the surge queue

September 21, 2020

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note: You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.

Note: Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

To flush a surge queue by using the CLI

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member (<`serverName`> and <`port`>) without specifying the name of the service group (<`name`>) and you cannot specify <`port`> without a <`serverName`>. Specify the <`serverName`> and <`port`> if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.

- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

Examples

```
flush ns surgeQ  - name SVC1ANZGB  - serverName 10.10.10.1 80
<!--NeedCopy-->
```

The above command flushes the surge queue of the service or virtual server that is named SVC1ANZGB and has IP address as 10.10.10

```
flush ns surgeQ
<!--NeedCopy-->
```

The above command flushes all the surge queues on the appliance.

To flush a surge queue by using the GUI

Navigate to **Traffic Management** > **Content Switching** > **Virtual Servers**, select a virtual server and, in the Action list, select **Flush Surge Queue**.

Manage a load balancing setup

September 21, 2020

An existing Load Balancing setup does not require a great deal of work to maintain as long as it is unchanged, but most do not remain unchanged for long. Increasing load requires new load-balanced servers and eventually new NetScaler appliances, which must be configured and added to the existing setup. Old servers wear out and need to be replaced, requiring removal of some servers and addition of others. Upgrades to your networking equipment or changes to topology may also require modifications to your load balancing setup. Therefore, you will need to perform operations on server objects, services, and virtual servers. The Visualizer can display your configuration graphically, and you can perform operations on the entities in the display. You can also take advantage of a number of other features that facilitate management of the traffic through your load balancing setup.

Manage server objects

September 21, 2020

During basic load balancing setup, when you create a service, a server object with the IP address of the service is created, if one does not already exist. If you prefer for your service objects named with domain names rather than IP addresses, you might also have created one or more server objects manually. You can enable, disable, or remove any server object.

When you enable or disable a server object, you enable or disable all services associated with the server object. When you refresh the NetScaler appliance after disabling a server object, the state of its service appears as OUT OF SERVICE. If you specify a wait time when disabling a server object, the server object continues to handle established connections for the specified amount of time, but rejects new connections. If you remove a server object, the service to which it is bound is also deleted.

To enable a server by using the CLI

At the command prompt, type:

```
enable server <name>
<!--NeedCopy-->
```

Example:

```
enable server 10.102.29.5
<!--NeedCopy-->
```

To enable or disable a server object by using the GUI

1. Navigate to Traffic Management > Load Balancing > Servers.
2. Select the server and, in the Action list, select Enable or Disable.

To disable a server object by using the CLI

At the command prompt, type:

```
disable server <name> <delay>
<!--NeedCopy-->
```

Example:

```
disable server 10.102.29.5 30
<!--NeedCopy-->
```

To remove a server object by using the CLI

At the command prompt, type:

```
rm server <name>
<!--NeedCopy-->
```

Example:

```
rm server 10.102.29.5
<!--NeedCopy-->
```

To remove a server object by using the GUI

1. Navigate to Traffic Management > Load Balancing > Servers.
2. Select a server, and click Remove.

Manage services

September 21, 2020

Services are enabled by default when you create them. You can disable or enable each service individually. When disabling a service, you normally specify a wait time during which the service continues to handle established connections, but rejects new ones, before shutting down. If you do not specify a wait time, the service shuts down immediately. During the wait time, the service's state is OUT OF SERVICE.

You can remove a service when it is no longer used. When you remove a service, it is unbound from its virtual server and deleted from the NetScaler configuration.

To enable or disable a service by using the command line interface

At the command prompt, type:

```
enable service <name>

disable service <name> <DelayInSeconds>
<!--NeedCopy-->
```

Examples:

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
<!--NeedCopy-->
```

To enable or disable a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service and, in the Action list, select Enable or Disable.

Manage a load balancing virtual server

September 21, 2020

Virtual servers are enabled by default when you create them. You can disable and enable virtual servers manually. If you disable a virtual server, the virtual service's state appears as OUT OF SERVICE. When this happens, the virtual server terminates all connections, either immediately or after allowing existing connections to complete, depending on the setting of the downStateFLush parameter. If downStateFLush is ENABLED (default), all the connections are flushed. If DISABLED, the virtual server continues to serve requests on existing connections.

You remove a virtual server only when you no longer require the virtual server. Before you remove it, you must unbind all services from it.

To enable or disable a virtual server by using the command line interface

At the command prompt, type:

```
enable lb vserver <name>
<!--NeedCopy-->
```

```
disable lb vserver <name>
<!--NeedCopy-->
```

Examples:

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
<!--NeedCopy-->
```

To enable or disable a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Select a virtual server, and in the Action list, select Enable or Disable.

To unbind a service from a virtual server by using the command line interface

At the command prompt, type:

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

To unbind a service from a virtual server by using the configuration utility

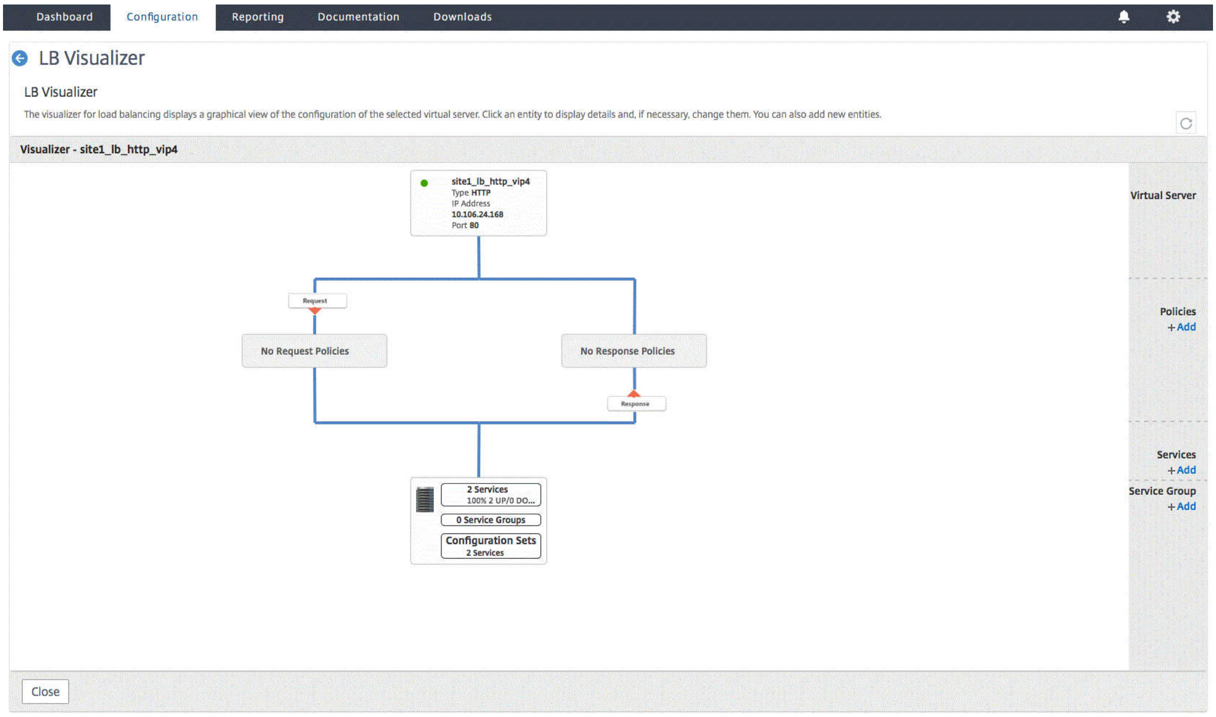
1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, and click in the Services section.
3. Select a service and click Unbind.

Load balancing visualizer

September 21, 2020

The Load Balancing Visualizer is a tool that you can use to view and modify the load balancing configuration in graphical format. Following is an example of the Visualizer display

Figure 1. Load Balancing Visualizer Display


Dashboard
Configuration
Reporting
Documentation
Downloads
LB Visualizer

LB Visualizer

The visualizer for load balancing displays a graphical view of the configuration of the selected virtual server. Click an entity to display details and, if necessary, change them. You can also add new entities.

Visualizer - site1_lb_http_vip4
site1_lb_http_vip4
Type HTTP
IP Address
10.106.24.168
Port 80
Request
No Request Policies
No Response Policies
Response
2 Services
100% 2 UP/0 DO...
0 Service Groups
Configuration Sets
2 Services
Virtual Server
Policies
+Add
Services
+Add
Service Group
+Add
Close

You can use the visualizer to view the following:

- The services and service groups that are bound to a virtual server.
- The monitors that are bound to each service.
- The policies that are bound to the virtual server.
- The policy labels, if configured.
- Configuration details of any displayed element.

You can also use the Visualizer to add and bind new objects, modify existing ones, and enable or disable objects. Most configuration elements displayed in the Visualizer appear under the same names as in other parts of the configuration utility. However, unlike the rest of the configuration utility, the Visualizer groups services that have the same configuration details and monitor bindings into an entity called a service container.

A service container is set of similar services and service groups that are bound to a single load balancing virtual server. The services in the container have the same properties, with the exception of the name, IP address, and port, and their monitor bindings should have the same weight and binding state. When you bind a new service to a virtual server, it is placed into an existing container if its configuration and monitor bindings match those of other services; otherwise, it is placed in its own container.

The following procedures provide only basic steps for using the Visualizer. Because the Visualizer duplicates functionality in other areas of the Load Balancing feature, other methods of viewing or configuring all of the settings that can be configured in the Visualizer are provided throughout the Load Balancing documentation.

Note: The Visualizer requires a graphic interface, so it is available only through the configuration utility.

To view load balancing virtual server properties by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.

To view configuration details for services, service groups, and monitors by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, double-click the entity to view the configuration details of the entity that is bound to this virtual server, you can do the following:

To view configuration details for policies and policy labels by using the Visualizer in the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to view, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, double-click policies entity to view policies that are bound to this virtual server.

To modify a resource in a load balancing configuration by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, on the Visualizer image, double-click the resource that you want to modify.

To add a load balancing configuration by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server that you want to configure, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, click **+** to add the resource.

Manage client traffic

September 21, 2020

Managing client connections properly helps to ensure that your applications remain available to users even when your NetScaler appliance is experiencing high loads. A number of load balancing features and other features available on the appliance can be integrated into a load balancing setup to process load more efficiently, divert it when necessary, and prioritize the tasks that the appliance must perform:

- **Sessionless load balancing.** You can configure sessionless load balancing virtual servers and perform load balancing without creating sessions in configurations that use DSR or intrusion detection systems (IDS).
- **Integrated caching.** You can redirect HTTP requests to a cache.
- **Priority queuing.** You can direct requests based on priority, by integrating your configuration with the Priority Queuing feature.
- **SureConnect.** You can use load balancing with the SureConnect feature to redirect important requests to a custom Web page, insulating them from delays due to network congestion.
- **Delayed cleanup.** You can configure delayed cleanup of virtual server connections to prevent the cleanup process from using CPU cycles during periods when the NetScaler appliance is experiencing high loads.
- **Rewrite.** You can use the Rewrite feature to modify port and protocol when performing HTTP redirection, or insert the virtual server IP address and port into a custom Request header.
- **RTSP NAT.**
- **Rate-based monitoring.** You can enable rate-based monitoring to divert excess traffic.
- **Layer 2 Parameters.** You can configure a virtual server to use the L2 parameters to identify a connection.
- **ICMP Response**. You can configure the appliance to send ICMP responses to PING requests according to your settings. On the IP address corresponding to the virtual server, set the ICMP RESPONSE to VSVR_CNTRLD, and on the virtual server, set the ICMP VSERVER RESPONSE.
  The following settings can be made on a virtual server:
  - When you set ICMP VSERVER RESPONSE to PASSIVE on all virtual servers, appliance always responds.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, appliance responds even if one virtual server is UP.
  - When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, appliance responds even if one virtual server set to ACTIVE is UP.

Configure sessionless load balancing virtual servers

September 21, 2020

When the NetScaler appliance performs load balancing, it creates and maintains sessions between clients and servers. The maintenance of session information places a significant load on the appliance resources, and sessions might not be needed in scenarios such as a direct server return (DSR) setup and the load balancing of intrusion detection systems (IDS). To avoid creating sessions when they are not necessary, you can configure a virtual server on the appliance for sessionless load balancing. In sessionless load balancing, the appliance carries out load balancing on a per-packet basis.

Sessionless load balancing can operate in MAC-based forwarding mode or IP-based forwarding mode.

For MAC-based forwarding, the IP address of the sessionless virtual server must be specified on all the physical servers to which the traffic is forwarded.

For IP-based forwarding in sessionless load balancing, the IP address and port of the virtual server need not be specified on the physical servers, because this information is included in the forwarded packets. When forwarding a packet from the client to the physical server, the appliance leaves client details such as IP address and port unchanged and adds the IP address and port of the destination.

Supported setup

NetScaler sessionless load balancing supports the following service types and load balancing methods:

Service Types

- ANY for MAC-based redirection
- ANY, DNS, and UDP for IP-based redirection

Load Balancing Methods

- Round Robin
- Least Bandwidth
- LRTM (Least response time method)
- Source IP Hash
- Destination IP Hash
- Source IP Destination IP Hash
- Source IP Source Port Hash
- Custom Load

Limitations

Sessionless load balancing has the following limitations:

- The appliance must be deployed in two-arm mode.
- A service must be bound to only one virtual server.
- Sessionless load balancing is not supported for service groups.
- Sessionless load balancing is not supported for domain based services (DBS services).
- Sessionless load balancing in the IP mode is not supported for a virtual server that is configured as a backup to a primary virtual server.
- You cannot enable spillover mode.
- For all the services bound to a sessionless load balancing virtual server, the Use Source IP (USIP) option must be enabled.
- For a wildcard virtual server or service, the destination IP address will not be changed.

Note:

- While configuring a virtual server for sessionless load balancing, explicitly specify a supported load balancing method. The default method, Least Connection, cannot be used for sessionless load balancing.

- To configure sessionless load balancing in MAC-based redirection mode on a virtual server, the MAC-based forwarding option must be enabled on the NetScaler appliance.

To add a sessionless virtual server by using the CLI

At the command prompt, type the following commands to add a sessionless virtual server and verify the configuration:

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -m <redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <load_balancing_method>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
 add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -lbMethod roundrobin -m ip
      Done
 show lb vserver sesslessv1
      sesslessv1 (11.11.12.123:54) - ANY Type: ADDRESS
      State: DOWN
      ...
      Effective State: DOWN
      Client Idle Timeout: 120 sec
      Down state flush: ENABLED
      ...
      Persistence: NONE
      Sessionless LB: ENABLED
      Connection Failover: DISABLED
      L2Conn: OFF
      1) Policy : cmp_text Priority:8680 Inherited
      2) Policy : cmp_nocmp_ie60 Priority:8690 Inherited
<!--NeedCopy-->
```

To configure sessionless load balancing on an existing virtual server

At the command prompt, type:

```
set lb vserver <name>@ -m <redirectionMode> -sessionless <(ENABLED|DISABLED)> -lbMethod <load_balancing_method>
<!--NeedCopy-->
```

Example

```
set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
  Done
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure a sessionless virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in Advanced Settings, click Traffic Settings, and then select Sessionless Load Balancing.

Redirect HTTP requests to a cache

September 21, 2020

The NetScaler cache redirection feature redirects HTTP requests to a cache. You can significantly reduce the impact of responding to HTTP requests and improve your Web site performance through proper implementation of the cache redirection feature.

A cache stores frequently requested HTTP content. When you configure cache redirection on a virtual server, the NetScaler appliance sends cacheable HTTP requests to the cache, and non-cacheable HTTP requests to the origin Web server.

To configure cache redirection on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -cacheable <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -cacheable yes
<!--NeedCopy-->
```

To configure cache redirection on a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Cacheable.

Direct requests according to priority

September 21, 2020

The NetScaler appliance supports prioritization of client requests with its priority queuing feature. This feature allows you to designate certain requests, such as those from important clients, as priority requests and sends them to the "front of the line," so that the appliance responds to them first. This allows you to provide uninterrupted service to those clients through demand surges or DDoS attacks on your web site.

To configure priority queuing on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -pq <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -pq yes
<!--NeedCopy-->
```

To configure priority queuing on a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Priority Queuing.

Note: You must configure priority queuing globally for it to function correctly.

Direct requests to a custom web page

September 21, 2020

The NetScaler appliance provides the SureConnect option to ensure that web applications respond despite delays caused by limited server capacity or processing speed. SureConnect does this by displaying an alternative web page of your choice when the server that hosts the primary web page is either unavailable or responding slowly.

To configure SureConnect on a virtual server, you must first configure the alternative content. For information about configuring a SureConnect website, see SureConnect. After you configure the website, enable SureConnect on the load balancing virtual server to put your SureConnect custom web page in use.

Note: For SureConnect to function correctly, you must configure it globally.

To enable SureConnect on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -sc <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -sc yes
<!--NeedCopy-->
```

To enable SureConnect on a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select SureConnect.

Enable cleanup of virtual server connections

September 21, 2020

Under certain conditions, you can configure the downStateFlush setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes and health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN.

A virtual server is marked DOWN only when all services bound to it are marked DOWN. When a virtual server goes DOWN, it terminates all connections, either immediately or after allowing existing connections to complete.

You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-TCP-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-TCP-1	State of connections
E	E	Both client and server connections are terminated.

Vserver-LB-1	Service-TCP-1	State of connections
E	D	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown

of a service, see
Graceful Shutdown of Services.

To configure the down state flush setting on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -downStateFlush <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
<!--NeedCopy-->
```

To configure the down state flush setting on a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click Traffic Settings, and select Down State Flush.

Rewrite ports and protocols for HTTP redirection

September 21, 2020

Virtual servers and the services that are bound to them may use different ports. When a service responds to an HTTP connection with a redirect, you might need to configure the NetScaler appliance to modify the port and the protocol to make sure that the redirection goes through successfully. You do this by enabling and configuring the redirectPortRewrite setting.

This setting affects only HTTP and HTTPS traffic. If this setting is enabled on a virtual server, the virtual server rewrites the port on redirects, replacing the port used by the service with the port used by the virtual server.

If the virtual server or service is of type SSL, you must enable SSL redirect on the virtual server or service. If both the virtual server and service are of type SSL, enable SSL redirect on the virtual server.

The redirectPortRewrite setting can be used in the following scenarios:

- The virtual server is of type HTTP and the services are of type SSL.
- The virtual server is of type SSL and the services are of type HTTP.
- The virtual server is of type HTTP and the services are of type HTTP.
- The virtual server is of type SSL and the services are of type SSL.

Scenario 1: The virtual server is of type HTTP and services are of type SSL. SSL redirect, and optionally port rewrite, is enabled on the service. If port rewrite is enabled, the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

Only SSL redirect is enabled. The virtual server can be configured on any port. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

SSL redirect and port rewrite are enabled. The virtual server is configured on port 80. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com/`

SSL redirect and port rewrite are enabled. Virtual server is configured on port 8080. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`http://domain.com:8080/`
`https://domain.com:444/`	`http://domain.com:8080/`

Scenario 2: The virtual server is of type SSL and services are of type HTTP. If port rewrite is enabled, only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

SSL redirect is enabled on the virtual server. The virtual server can be configured on any port. See the *following table.*

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`https://domain.com/`
`http://domain.com:8080/`	`https://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`https://domain.com/`
`http://domain.com:8080/`	`https://domain.com/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

SSL redirect and port rewrite are enabled. The virtual server is configured on port 444. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`https://domain.com:444/`
`http://domain.com:8080/`	`https://domain.com:444/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:445/`	`https://domain.com:445/`

Scenario 3: The virtual server and service are of type HTTP. Port rewrite must be enabled on the virtual server. Only the port of HTTP URLs is rewritten. HTTPS URLs from the server are sent as is to the client.

The virtual server is configured on port 80. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

The virtual server is configured on port 8080. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com:8080/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:445/`	`https://domain.com:445/`

Scenario 4: The virtual server and service are of type SSL. If port rewrite is enabled, only the port of HTTPS URLs is rewritten. HTTP URLs from the server are sent as is to the client.

SSL redirect is enabled on the virtual server. The virtual server can be configured on any port. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com:444/`

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 443. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`

Redirect URL from the Server	Redirect URL sent to the Client
`https://domain.com/`	`https://domain.com/`
`https://domain.com:444/`	`https://domain.com/`

SSL redirect and port rewrite are enabled on the virtual server. The virtual server is configured on port 444. See the following table:

Redirect URL from the Server	Redirect URL sent to the Client
`http://domain.com/`	`http://domain.com/`
`http://domain.com:8080/`	`http://domain.com:8080/`
`https://domain.com/`	`https://domain.com:444/`
`https://domain.com:445/`	`https://domain.com:444/`

To configure HTTP redirection on a virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name> -redirectPortRewrite (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
<!--NeedCopy-->
```

To configure HTTP redirection on a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in the Advanced Settings pane, click Traffic Settings, and then select Rewrite.

To configure SSL Redirect on an SSL virtual server or service by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)

set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled

set ssl service service-SSL-1 -sslRedirect enabled
<!--NeedCopy-->
```

To configure SSL redirection and SSL port rewrite on an SSL virtual server or service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open the virtual server.
2. In Advanced Settings, click SSL Parameters, and select SSL Redirect.

Insert IP address and port of a virtual server in the request header

September 21, 2020

If you have multiple virtual servers that communicate with different applications on the same service, you must configure the NetScaler appliance to add the IP address and port number of the appropriate virtual server to the HTTP requests that are sent to that service. This setting allows applications running on the service to identify the virtual server that sent the request.

If the primary virtual server is down and the backup virtual server is up, the configuration settings of the backup virtual server are added to the client requests. If you want the same header tag to be added, regardless of whether the requests are from the primary virtual server or backup virtual server, then you must configure the required header tag on both virtual servers.

Note: This option is not supported for wild card virtual servers or dummy virtual servers.

To insert the IP address and port of the virtual server in the client requests by using the CLI

At the command prompt, type:

```
set lb vserver <name> -insertVserverIPPort <insertVserverIPPort> [<
    vipHeader>]
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
<!--NeedCopy-->
```

To insert the IP address and port of the virtual server in the client requests by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open the virtual server, and in the Advanced Settings pane, click Traffic Settings, and then select Virtual Server IP Port Insertion and specify a virtual server IP port header.

Use a specified source IP for backend communication

September 21, 2020

For communication with the physical servers or other peer devices, the NetScaler appliance uses an IP address owned by it as the source IP address. The NetScaler appliance maintains a pool of its IP addresses, and dynamically selects an IP address while connecting with a server. Depending on the subnet in which the physical server is placed, the appliance decides which IP address to use. This address pool is used for sending traffic as well as monitor probes.

In many situations, you may want the appliance to use a specific IP address or any IP address from a specific set of IP addresses for backend communications. The following are a few examples:

- A server can distinguish monitor probes from traffic if the source IP address used for monitor probes belongs to a specific set.
- To improve server security, a server may be configured to respond to requests from a specific set of IP addresses or, sometimes, from a single specific IP address. In such a case, the appliance can use only the IP addresses accepted by the server as the source IP address.
- The appliance can manage its internal connections efficiently if it can distribute its IP addresses into IP sets and use an address from a set only for connecting to a specific service.

To configure the appliance to use a specified source IP address, create net profiles (network profiles) and configure the appliance entities to use the profile. A net profile can be bound to load balancing or content switching virtual servers, services, service groups, or monitors. A net profile has NetScaler owned IP addresses (SNIPs and VIPs) that can be used as the source IP address. It can be a single IP address or a set of IP addresses, referred to as an IP set. If a net profile has an IP set, the appliance dynamically selects an IP address from the IP set at the time of connection. If a profile has a single IP address, the same IP address is used as the source IP.

If a net profile is bound to a load balancing or content switching virtual server, the profile will be used for sending traffic to all the services bound to it. If a net profile is bound to a service group, the appliance uses the profile for all the members of the service group. If a net profile is bound to a monitor, the appliance uses the profile for all the probes sent from the monitor.

Note: When a NetScaler appliance uses a VIP address to communicate with a server, it uses session entries to identify whether the traffic destined to the VIP address is a response from a server or a request from a client.

Usage of a net profile for sending traffic

If the Use Source IP Address (USIP) option is enabled, the appliance uses the IP address of the client and ignores all the net profiles. If the USIP option is not enabled, the appliance selects the source IP in the following manner:

- If there is no net profile on the virtual server or the service/service group, the appliance uses the default method.
- If there is a net profile only on the service/service group, the appliance uses that net profile.
- If there is a net profile only on the virtual server, the appliance uses the net profile.
- If there is a net profile both on the virtual server and service/service group, the appliance uses the net profile bound to the service/service group.

Usage of a net profile for sending monitor probes:

For monitor probes, the appliance selects the source IP in the following manner:

- If there is a net profile bound to the monitor, the appliance uses the net profile of the monitor. It ignores the net profiles bound to the virtual server or service/service group.
- If there is no net profile bound to the monitor,
  - If there is a net profile on the service/service group, the appliance uses the net profile of the service/service group.
  - If there is no net profile even on the service/service group, the appliance uses the default method of selecting a source IP.

Note: If there is no net profile bound to a service, the appliance looks for a net profile on the service group if the service is bound to a service group.

To use a specified source IP address for communication, go through the following steps:

1. Create IP sets from the pool of SNIPs and VIPs owned by the NetScaler appliance. An IP set can consist of both SNIP and VIP addresses. For instructions, see Creating IP Sets.
2. Create net profiles. For instructions, see Creating a Net Profile.
3. Bind the net profiles to the appliance entities. For instructions, see Binding a Net Profile to a NetScaler Entity.

> **Note**:
>
> - A net profile can have only the IP addresses specified as SNIP and VIP on the NetScale ADC appliance.
> - Source IP persistence is not honored for NetScaler initiated packets.

Manage net profiles

A net profile (or network profile) contains an IP address or an IP set. During communication with physical servers or peers, the NetScaler appliance uses the addresses specified in the profile as the source IP address.

- For instructions on creating a network profile, see Creating a Network Profile.
- For instructions on binding a network profile to a NetScaler entity, see Binding a Net Profile to a NetScaler Entity.

Create an IP set

An IP set is a set of IP addresses, which are configured on the NetScaler appliance as Subnet IP addresses (SNIPs) or Virtual IP addresses (VIPs). An IP set is identified with a meaningful name that helps in identifying the usage of the IP addresses contained in it. To create an IP set, add an IP set and bind NetScaler owned IP addresses to it. SNIP addresses and VIP addresses can be present in the same IP set.

To create an IP set by using the CLI

At the command prompt, type the following commands:

```
add ipset <name>

bind ipset <name> <IPAddress>
<!--NeedCopy-->
```

or

```
bind ipset <name> <IPAddress>

show ipset [<name>]
<!--NeedCopy-->
```

The above command shows the names of all the IP sets on the appliance if you do not pass any name. It shows the IP addresses bound to the specified IP set if you pass a name.

Examples

```
1.
> add ipset skpnwipset
 Done
> bind ipset skpnwipset 21.21.20.1
 Done

2.
 > add ipset testnwipset
 Done
> bind ipset testnwipset 21.21.21.[21-25]
 IPAddress "21.21.21.21" bound
 IPAddress "21.21.21.22" bound
 IPAddress "21.21.21.23" bound
 IPAddress "21.21.21.24" bound
 IPAddress "21.21.21.25" bound
 Done

3.
 > bind ipset skpipset 11.11.11.101
 ERROR: Invalid IP address
[This IP address could not be added because this is not an IP address
   owned by the NetScaler appliance]
 > add ns ip 11.11.11.101 255.255.255.0 -type SNIP
 ip "11.11.11.101" added
 Done
 > bind ipset skpipset 11.11.11.101
 IPAddress "11.11.11.101" bound
 Done
4.
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
 Done

5.
> sh ipset skpnewipset
IP:21.21.21.21
IP:21.21.21.22
IP:21.21.21.23
IP:21.21.21.24
IP:21.21.21.25
```

```
 Done
<!--NeedCopy-->
```

To create an IP set by using the GUI

Navigate to System > Network > IP Sets, and create an IP set.

Create a net profile

A net profile (network profile) consists of one or more SNIP
or VIP addresses of the NetScaler appliance.

To create a net profile by using the CLI

At the command prompt, type:

```
add netprofile <name> [-srcIp <srcIpVal>]
<!--NeedCopy-->
```

If the srcIpVal is not provided in this command, it can be provided later by using the
set netprofile command.

Examples

```
add netprofile skpnetprofile1 -srcIp 21.21.20.1
Done

add netprofile baksnp -srcIp bakipset
Done

set netprofile yahnp -srcIp 12.12.23.1
Done

set netprofile citkbnp -srcIp citkbipset
Done
<!--NeedCopy-->
```

Bind a net profile to a NetScaler entity

A net profile can be bound to a load balancing virtual server, service, service group, or a monitor.

Note: You can bind a net profile at the time of creating a NetScaler entity or bind it to an existing entity.

To bind a net profile to a server by using the command line interface

You can bind a net profile to load balancing virtual servers and content switching virtual servers. Specify the appropriate virtual server.

At the command prompt, type:

```
set lb vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

or

```
set cs vserver <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Examples

```
set lb vserver skpnwvs1 -netProfile gntnp
 Done
set cs vserver mmdcsv -netProfile mmdnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open the virtual server.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a service by using the CLI

At the command prompt, type:

```
set service <name> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Example

```
set service brnssvc1 -netProfile brnsnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a service by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Services**, and open a service.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a service group by using the CLI

At the command prompt, type:

```
set servicegroup <serviceGroupName> -netProfile <net_profile_name>
<!--NeedCopy-->
```

Example

```
set servicegroup ndhsvcgrp -netProfile ndhnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a service group by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Service Groups**, and open a service group.
2. In Advanced Settings, click **Profiles**, and set a net profile.

To bind a net profile to a monitor by using the CLI

At the command prompt, type:

set monitor -netProfile

Example

```
set monitor brnsecvmon1 -netProfile brnsmonnp
 Done
<!--NeedCopy-->
```

To bind a net profile to a monitor by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Open a monitor, and set the net profile.

Set a time-out value for idle client connections

April 6, 2021

You can configure a virtual server to terminate any idle client connections after a configured time-out period (in seconds) elapses. When you configure this setting, the NetScaler appliance waits for the time you specify and, if the client is idle after that time, it closes the client connection. By default, the client idle time-out value is set to 180 seconds.

To set a time-out value for idle client connections by using the CLI

At the command prompt, type:

```
set lb vserver <name> -cltTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -cltTimeout 100
<!--NeedCopy-->
```

To set a time-out value for idle client connections by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In **Advanced Settings,** click **Traffic Settings**, and set the client idle time-out value in seconds.

Manage RTSP connections

September 21, 2020

The NetScaler appliance can use either of two topologies—NAT-on mode or NAT-off mode—to load balance RTSP servers. In NAT-on mode, Network Address Translation (NAT) is enabled and configured on the appliance. RTSP requests and responses both pass through the appliance. You must therefore configure the appliance to perform network address translation (NAT) to identify the data connection.

For more information about enabling and configuring NAT, see "IP Addressing."

In NAT-off mode, NAT is not enabled and configured. The appliance receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. The load balanced RTSP servers send their responses directly to the client, bypassing the appliance. You must therefore configure the appliance to use Direct Server Return (DSR) mode, and assign publicly accessible FQDNs in DNS to your load balanced RTSP servers.

For more information about enabling and configuring DSR mode, see "Configuring Load Balancing in Direct Server Return Mode." For more information about configuring DNS, see "Domain Name System. In either case, when you configure RTSP load balancing, you must also configure rtspNat to match the topology of your load balancing setup.

To configure RTSP NAT by using the CLI

At the command prompt, type:

```
set lb vserver <name>  - RTSPNAT <ValueOfRTSPNAT>
<!--NeedCopy-->
```

Example:

```
set lb vserver vserver-LB-1  - RTSPNAT ON
<!--NeedCopy-->
```

To configure RTSP NAT by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server of type RTSP.
2. In Advanced Settings, click **Traffic Settings**, and select **RTSP Natting**.

Manage client traffic on the basis of traffic rate

September 21, 2020

You can monitor the rate of traffic that flows through load balancing virtual servers and control the behavior of the NetScaler appliance based on the traffic rate. You can throttle the traffic flow if it is too high, cache information based on the traffic rate, and if the traffic rate is too high redirect excess traffic to a different load balancing virtual server. You can apply rate-based monitoring to HTTP and Domain Name System (DNS) requests.

For more information on rate-based policies, see Rate Limiting.

Identify a connection with layer 2 parameters

September 21, 2020

Generally, to identify a connection, NetScaler appliance uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.

Enabling the L2Conn parameter for a load balancing virtual server allows multiple TCP and non-TCP connections with the same 4-tuple (<`source IP`>:<`source port`>::<`destination IP`>:<`destination port`>) to co-exist on the NetScaler appliance. The appliance uses both the 4-tuple and the Layer 2 parameters to identify TCP and non-TCP connections.

You can enable the L2Conn option in the following scenarios:

- Multiple VLANs are configured on the NetScaler appliance, and a firewall is set up for each VLAN.
- You want the traffic originating from the servers in one VLAN and bound for a virtual server in another VLAN to pass through the firewalls configured for both VLANs.

Therefore, when an nCore NetScaler appliance on which the l2Conn parameter is set for one or more load balancing virtual servers is downgraded to a Classic build or to an nCore build that does not support the l2Conn parameter, the load balancing configurations that use the l2Conn parameter become ineffective.

To configure the L2 connection option by using the CLI

At the command prompt, type:

```
add lb vserver <name> <serviceType> <IPAddress>@ <port> -l2Conn ON
<!--NeedCopy-->
```

Example

```
 add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
<!--NeedCopy-->
```

To configure the L2 connection option by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Layer 2 Parameters.

Configure the prefer direct route option

September 21, 2020

On a wildcard load balancing virtual server if you explicitly configure a route to a destination, by default, the NetScaler appliance forwards traffic according to the configured route. If you want the appliance to not look up for the configured route, you can set the Prefer Direct Route option to NO.

If a device is directly connected to a NetScaler appliance, the appliance directly forwards traffic to the device. For example, if the destination of a packet is a firewall, the packet need not be routed through another firewall. However, in some cases, you may want the traffic to go through the firewall even if the device is directly connected to it. In such cases, you can set the Prefer Direct Route Option to NO.

Note: The preferDirectRoute setting is applicable to all the wildcard virtual servers on the NetScaler appliance.

To set the prefer direct route option by using the CLI

At the command prompt, type:

```
set lb parameter -preferDirectRoute (YES | NO)
<!--NeedCopy-->
```

Example:

```
set lb parameter -preferDirectRoute YES
<!--NeedCopy-->
```

To set the prefer direct route option by using the GUI

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing parameters.
2. Select Prefer Direct Route.

Use a source port from a specified port range for backend communication

September 21, 2020

By default, for configurations with USIP option disabled or with USIP and use proxy port options enabled, the NetScaler appliance communicates to the servers from a random source port (greater than 1024).

The appliance supports using a source port from a specified port range for communicating to the servers. One of the use case of this feature is for servers that are configured to identify received traffic belonging to a specific set on the basis of source port for logging and monitoring purposes. For example, identifying internal and external traffic for logging purpose.

Configuring the NetScaler appliance to use a source port from a port range for communicating to the servers consists of the following tasks:

- **Create a net profile and set the source port range parameter**. A source port range parameter specifies one or more port ranges. The appliance randomly selects one of the free ports from the specified port ranges and used it as the source port for each connection to servers.
- **Bind the net profile to load balancing virtual servers, services, or service groups**: A net profile with source port range setting can be bound to a virtual server, service, or a service group of a load balancing configuration. For a connection to a virtual server, the appliance randomly selects one of the free ports from the specified port ranges of a net profile and use this port as the source port for connecting to one of the bound server.

To specify a source port range or ranges by using the CLI

At the command prompt, type:

```
bind netProfile <name> (-srcPortRange <int[-int]> ...)

show netprofile <name>
<!--NeedCopy-->
```

To specify a source port range or ranges by using the GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the **Source Port Range** parameter while adding or modifying NetProfiles.

Sample Configuration

In the following sample configuration, net profile PARTIAL-NAT-1 has partial NAT settings and is bound to load balancing virtual server LBVS-1, which is of type ANY. For packets received on LBVS-1 from 192.0.0.0/8, the NetScaler appliance translates the last octet of the packet's source IP address to 100. For example, a packet with source IP address 192.0.2.30 received on LBVS-1, the NetScaler appliance translates the source IP address to 100.0.2.30 before sending it one of the bound servers.

```
> add netprofile CUSTOM-SRCPORT-NP-1
Done
> bind netprofile CUSTOM-SRCPRT-NP-1  - srcportrange 2000-3000
```

```

Done
> bind netprofile CUSTOM-SRCPRT-NP-1  – srcportrange 5000-6000

Done
> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1

Done
```

Configure source IP persistency for backend communication

September 21, 2020

By default, for a load balancing configuration with the USIP option disabled and a net profile bound to a virtual server or services or service groups, the NetScaler appliance uses the round robin algorithm to select an IP address from the net profile for communicating with the servers. Because of this selection method, the IP address selected can be different for different sessions of a specific client.

Some situations require that the NetScaler appliance send all of a specific client's traffic from the same IP address when sending the traffic to servers. The servers can then, for example, identify traffic belonging to a specific set for logging and monitoring purposes.

The source IP persistency option of a net profile enables the NetScaler appliance to use the same address, specified in the net profile, to communicate with servers about all sessions initiated from a specific client to a virtual server.

To enable source IP persistency in a net profile by using the CLI

To enable source IP persistency while adding a net profile, at the command prompt, type:

```
add netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable source IP persistency in an existing net profile, at the command prompt, type:

```
set netProfile <name> -srcippersistency ( ENABLED | DISABLED )

show netprofile <name>
<!--NeedCopy-->
```

To enable source IP persistency in a net profile by using the GUI

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Select **Source IP Persistency** while adding or modifying a net profile.

Example

In the following sample configuration, net profile NETPROFILE-IPPRSTNCY-1 has the source IP persistency option enabled and is bound to load balancing virtual server LBVS-1.

The NetScaler appliance always use the same IP address (in this example, 192.0.2.11) to communicate with servers bound to LBVS-1, for all sessions initiated from a specific client to the virtual server.

```
> add ipset IPSET-1

Done
> bind ipset IPSET-1 192.0.2.[11-15]
 IPAddress "192.0.2.11" bound
 IPAddress "192.0.2.12" bound
 IPAddress "192.0.2.13" bound
 IPAddress "192.0.2.14" bound
 IPAddress "192.0.2.15" bound
 Done
> add netprofile NETPROFILE-IPPRSTNCY-1 -srcIp IPSET-1 -
   srcippersistency ENABLED

Done
> set lb vserver LBVS-1 -netprofile NETPROFILE-IPPRSTNCY-1

 Done
```

Advanced load balancing settings

September 21, 2020

In addition to configuring virtual servers, you can configure advanced settings for services.

To configure advanced load balancing settings, see the following sections:

- Gradually Stepping Up the Load on a New Service with Virtual Server-Level Slow Start
- The No-Monitor Option for Services
- Protecting Applications on Protected Servers Against Traffic Surges
- Enabling Cleanup of Service Connections

- Directing Requests to a Custom Web Page
- Enabling Access to Services When Down
- Enabling TCP Buffering of Responses
- Enabling Compression
- Maintaining Client Connection for Multiple Client Requests
- Inserting the IP Address of the Client in the Request Header
- Using the Source IP Address of the Client When Connecting to the Server
- Configuring the Source Port for Server-Side Connections
- Setting a Limit on the Number of Client Connections
- Setting a Limit on Number of Requests Per Connection to the Server
- Setting a Threshold Value for the Monitors Bound to a Service
- Setting a Timeout Value for Idle Client Connections
- Setting a Timeout Value for Idle Server Connections
- Setting a Limit on the Bandwidth Usage by Clients
- Redirecting Client Requests to a Cache
- Retaining the VLAN Identifier for VLAN Transparency
- Configuring Automatic State Transition Based on Percentage Health of Bound Services

Gradually stepping up the load on a new service with virtual server–level slow start

September 21, 2020

You can configure the NetScaler appliance to gradually increase the load on a service (the number of requests that the service receives per second) immediately after the service is either added to a load balancing configuration or has a state change from DOWN to UP (hereafter, the term "new service" is used for both situations). You can either increase the load manually with load values and intervals of your choice (manual slow start) or configure the appliance to increase the load at a specified interval (automated slow start) until the service is receiving as many requests as the other services in the configuration. During the ramp-up period for the new service, the appliance uses the configured load balancing method.

This functionality is not available globally. It has to be configured for each virtual server. The functionality is available only for virtual servers that use one of the following load balancing methods:

- Round robin
- Least connection
- Least response time
- Least bandwidth
- Least packets

- LRTM (Least Response Time Method)
- Custom load

For this functionality, you need to set the following parameters:

- The new service request rate, which is the amount by which to increase the number or percentage of requests sent to a new service each time the rate is incremented. That is, you specify the size of the increment in terms of either the number of requests per second or the percentage of the load being borne, at the time, by the existing services. If this value is set to 0 (zero), slow start is not performed on new services.

  Note: In automated slow start mode, the final increment is smaller than the specified value if the specified value would place a heavier load on the new service than on the other services.

- The increment interval, in seconds. If this value is set to 0 (zero), the load is not incremented automatically. You have to increment it manually.

With automated slow start, a service is taken out of the slow start phase when one of the following conditions applies:

- The actual request rate is less than the new service request rate.
- The service does not receive traffic for three successive increment intervals.
- The request rate has been incremented 200 times.
- The percentage of traffic that the new service must receive is greater than or equal to 100.

With manual slow start, the service remains in the slow start phase until you take it out of that phase.

Manual slow start

If you want to manually increase the load on a new service, do not specify an increment interval for the load balancing virtual server. Specify only the new service request rate and the units. With no interval specified, the appliance does not increment the load periodically. It maintains the load on the new service at the value specified by the combination of the new service request rate and units until you manually modify either parameter. For example, if you set the new service request rate and unit parameters to 25 and "per second," respectively, the appliance maintains the load on the new service at 25 requests per second until you change either parameter. When you want the new service to exit the slow start mode and receive as many requests as the existing services, set the new service request rate parameter to 0.

As an example, assume that you are using a virtual server to load balance 2 services, Service1 and Service2, in round robin mode. Further assume that the virtual server is receiving 240 requests per second, and that it is distributing the load evenly across the services. When a new service, Service3, is added to the configuration, you might want to increase the load on it manually through values of 10, 20, and 40 requests per second before sending it its full share of the load. The following table shows the values to which you set the three parameters.

Table 1. Parameter Values

Parameter	Value
Interval in seconds	0
New service request rate	10, 20, 40, and 0, at intervals that you choose
Units for the new service request rate	Requests per second

When you set the new service request rate parameter to 0, Service3 is no longer considered a new service, and receives its full share of the load.

Assume that you add another service, Service4, during the ramp-up period for Service3. In this example, Service4 is added when the new service request rate parameter is set to 40. Therefore, Service4 begins receiving 40 requests per second.

The following table shows the load distribution on the services during the period described in this example.

Table 2. Load Distribution on Services when Manually Stepping Up the Load

	new service request rate = 10 req/sec (Service3added)	new service request rate = 20 req/sec	new service request rate = 40 req/sec (Service4added)	new service request rate = 0 req/sec (new services exit slow start mode)
Service1	115	110	80	60
Service2	115	110	80	60
Service3	10	20	40	60
Service4	-	-	40	60
Total req/sec (load on the virtual server)	240	240	240	240

Automated slow start

If you want the appliance to increase the load on a new service automatically at specified intervals until the service can be considered capable of handling its full share of the load, set the new service request rate parameter, the units parameter, and the increment interval. When all the parameters are set to values other than 0, the appliance increments the load on a new service by the value of the new service request rate, at the specified interval, until the service is receiving it's full share of the load.

As an example, assume that four services, Service1, Service2, Service3, and Service4, are bound to a load balancing virtual server, vserver1. Further assume that vserver1 receives 100 requests per second, and that it distributes the load evenly across the services (25 requests per second per service). When you add a fifth service, Service5, to the configuration, you might want the appliance to send the new service 4 requests per second for the first 10 seconds, 8 requests per second for the next 10 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters:

Table 3. Parameter Values

Parameter	**Value**
Interval in seconds	10
Increment value	4
Units for the new service request rate	Requests per second

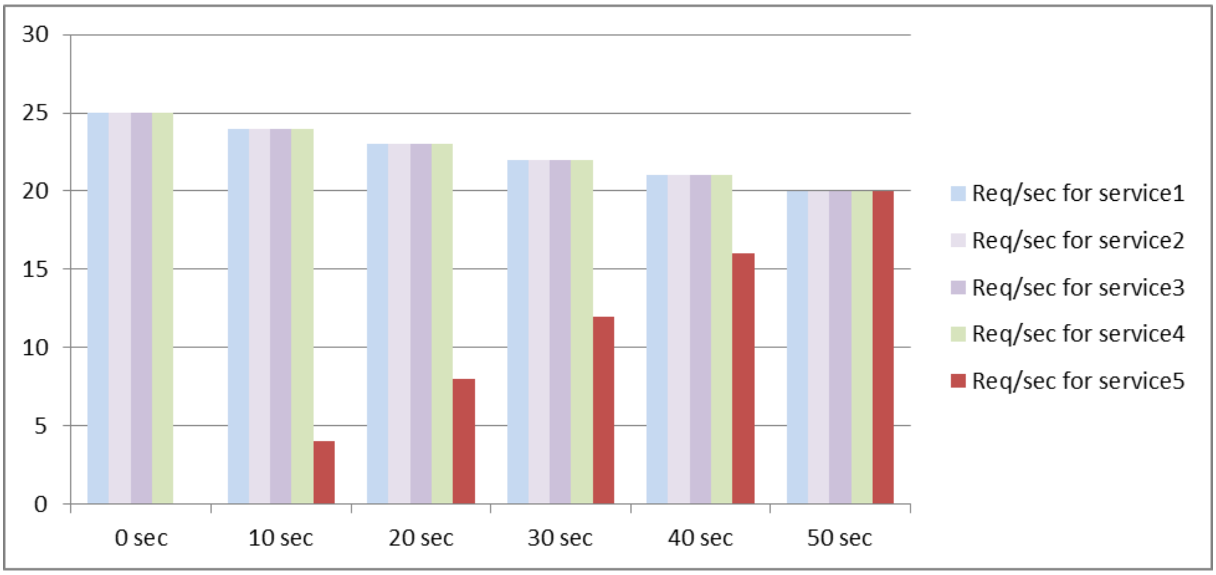
With this configuration, the new service begins receiving as many requests as the existing services 50 seconds after it is added or its state has changed from DOWN to UP. During each interval in this period, the appliance distributes to the existing servers the excess of requests that would have been sent to the new service in the absence of stepwise increments. For example, in the absence of stepwise increments, each service, including Service5, would have received 20 requests each per second. With stepwise increments, during the first 10 seconds, when Service5 receives only 4 requests per second, the appliance distributes the excess of 16 requests per second to the existing services, resulting in the distribution pattern shown in the following table and figure over the 50-second period. After the 50-second period, Service5 is no longer considered a new service, and it receives its normal share of traffic.

Table 4. Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added

	0 sec	10 sec	20 sec	30 sec	40 sec	50 sec
Req/sec forService1	25	24	23	22	21	20
Req/sec forService2	25	24	23	22	21	20

	0 sec	10 sec	20 sec	30 sec	40 sec	50 sec
Req/sec forService3	25	24	23	22	21	20
Req/sec forService4	25	24	23	22	21	20
Req/sec forService5	0	4	8	12	16	20
Total req/sec (load on the virtual server)	100	100	100	100	100	100

Figure 1. A Graph of the Load Distribution Pattern on All Services for the 50-second Period Immediately after Service5 is Added



An alternative requirement might be for the appliance to send Service5 25% of the load on the existing services in the first 5 seconds, 50% in the next 5 seconds, and so on, until it is receiving 20 requests per second. For this requirement, the following table shows the values to which you set the three parameters.

Table 5. Parameter Values

Parameter	Value
Interval in seconds	5
Increment value	25
Units for the new service request rate	Percent

With this configuration, the service begins receiving as many requests as the existing services 20 seconds after it is added or its state has changed from DOWN to UP. The traffic distribution during the ramp-up period for the new service is identical to the one described earlier, where the unit for the step increments was "requests per second."

Set the slow start parameters

You set the slow start parameters by using either the set lb vserver or the add lb vserver command. The following command is for setting slow start parameters when adding a virtual server.

To configure stepwise load increments for a new service by using the command line interface

At the command prompt, type the following commands to configure stepwise increments in the load for a service and verify the configuration:

```
add lb vserver <name> <serviceType> <IPAddress> <port> [-
    newServiceRequest <positive_integer>] [<newServiceRequestUnit>] [-
    newServiceRequestIncrementInterval <positive_integer>]

show lb vserver <name>
<!--NeedCopy-->
```

Example

```
set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -
    newServiceRequestIncrementInterval 10
Done

show lb vserver BR_LB
BR_LB (192.0.2.33:80) - HTTP Type: ADDRESS
State: UP
...
...
New Service Startup Request Rate: 5 PER_SECOND, Increment Interval: 10
```

```
...
...
Done
<!--NeedCopy-->
```

To configure stepwise load increments for a new service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Method, and set the following slow start parameters:
    - New Service Startup Request Rate.
    - New Service Request Unit.
    - Increment Interval.

The no-monitor option for services

September 21, 2020

If you use an external system to perform health checks on the services and do not want the NetScaler appliance to monitor the health of a service, you can set the no-monitor option for the service. If you do so, the appliance does not send probes to check the health of the service but shows the service as UP. Even if the service goes DOWN, the appliance continues to send traffic from the client to the service as specified by the load balancing method.

The monitor can be in the ENABLED or DISABLED state when you set the no-monitor option, and when you remove the no-monitor option, the earlier state of the monitor is resumed.

You can set the no-monitor option for a service when creating the service. You can also set the no-monitor option on an existing service.

The following are the consequences of setting the no-monitor option:

- If a service for which you enabled the no-monitor option goes down, the appliance continues to show the service as UP and continues to forward traffic to the service. A persistent connection to the service can worsen the situation. In that case, or if many services shown as UP are actually DOWN, the system may fail. To avoid such a situation, when the external mechanism that monitors the services reports that a service that is DOWN, remove the service from the NetScaler configuration.
- If you configure the no-monitor option on a service, you cannot configure load balancing in the Direct Server Return (DSR) mode. For an existing service, if you set the no-monitor option, you cannot configure the DSR mode for the service.

To set the no-monitor option for a new service by using the CLI

At the command prompt, type the following commands to create a service with the health monitor option, and verify the configuration:

```
add service <serviceName> <IP | serverName> <serviceType> <port> -
    healthMonitor (YES|NO)
<!--NeedCopy-->
```

Example:

```
add service nomonsrvc 10.102.21.21 http 80 -healthMonitor no
Done

show service nomonsrvc
nomonsrvc (10.102.21.21:80) - HTTP
State: UP
Last state change was at Mon Nov 15 22:41:29 2010
Time since last state change: 0 days, 00:00:00.970
Server Name: 10.102.21.21
 Server ID : 0 Monitor Threshold : 0
...
Access Down Service: NO
...
Down state flush: ENABLED
Health monitoring: OFF

1 bound monitor:
1) Monitor Name: tcp-default
State: UNKNOWN Weight: 1
Probes: 3 Failed [Total: 3 Current: 3]
Last response: Probe skipped - Health monitoring is turned off.
Response Time: N/A
Done
<!--NeedCopy-->
```

To set the no-monitor option for an existing service by using the CLI

At the command prompt, type the following command to set the health monitor option:

```
set service <name> -healthMonitor (YES|NO)
<!--NeedCopy-->
```

Example:

```
By default, the state of a service and the state of the corresponding monitor are UP.
 >show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP


1) Monitor Name: http-ecv
 State: UP Weight: 1
Probes: 99992 Failed [Total: 0 Current: 0]
Last response: Success - Pattern found in response.
Response Time: 3.76 millisec
Done

When the no-monitor option is set on a service, the state of the monitor changes to UNKNOWN.
set service LB-SVC1 -healthMonitor NO
Done

show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010.
Time since last state change: 5 days, 18:55:48.710
Health monitoring: OFF

1) Monitor Name: http-ecv
   State: UNKNOWN Weight: 1
         Probes: 100028 Failed [Total: 0 Current: 0]
   Last response: Probe skipped - Health monitoring is turned off.
     Response Time: 0.0 millisec
  Done
When the no-monitor option is removed, the earlier state of the monitor is resumed.
> set service LB-SVC1 -healthMonitor YES
Done
>show service LB-SVC1
LB-SVC1 (10.102.29.5:80) - HTTP
State: UP
Last state change was at Fri Dec 10 10:17:37 2010
Time since last state change: 5 days, 18:57:47.880
1) Monitor Name: http-ecv
   State: UP Weight: 1
   Probes: 100029 Failed [Total: 0 Current: 0]
```

```
    Last response: Success - Pattern found in response.
    Response Time: 5.690 millisec
    Done
<!--NeedCopy-->
```

To set the no-monitor option for a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and clear Health Monitoring.

Protect applications on protected servers against traffic surges

September 21, 2020

The NetScaler appliance provides the surge protection option to maintain the capacity of a server or cache. The appliance regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The appliance blocks any surges passed to the server, thereby preventing overloading of the server.

For surge protection to function correctly, you must enable it globally. For more information about surge protection, see "Surge Protection."

To set surge protection on the service by using the CLI

At the command prompt, type:

```
set service <name> -sp <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -sp ON
<!--NeedCopy-->
```

To set surge protection on the service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a source.
2. In Advanced Settings, select Traffic Settings, and select Surge Protection.

Enable cleanup of virtual server and service connections

September 21, 2020

The state of a virtual server depends on the states of the services bound to it. The state of each service depends on the responses of the load balanced servers to probes or health checks sent by the monitors that are bound to that service. Sometimes the load balanced servers do not respond. If a server is slow or busy, monitoring probes can time out. If repeated monitoring probes are not answered within the configured timeout period, the service is marked DOWN. If a service or virtual server are marked DOWN, the server and client side connections must be flushed. Terminating existing connections frees resources, and in certain cases speeds recovery of overloaded load balancing setups.

Under certain conditions, you can configure the **downStateFlush** setting to immediately terminate existing connections when a service or a virtual server is marked DOWN. You must not enable the downStateFlush setting on those application servers that must complete their transactions. You can enable this setting on Web servers whose connections can safely be terminated when they marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, Vserver-LB-1, with one service bound to it, Service-1. In the table, E and D denote the state of the downStateFlush setting: E means Enabled, and D means Disabled.

Vserver-LB-1	Service-1	State of Connections
E	E	Both client and server connections are terminated.

Vserver-LB-1	Service-1	State of Connections
E	D	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only client connections are terminated.
D	E	For some service types, such as TCP, for which the NetScaler appliance does not support connection reuse, both client and server connections are terminated. For service types, such as HTTP, for which the appliance supports connection reuse, both client and server connections are terminated only if a transaction is active on those connections. If a transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

If you want to disable a service only when all the established connections are closed by the server or the client, you can use the graceful shutdown option. For information about the graceful shutdown

of a service, see
Graceful Shutdown of Services.

To set down state flush on the service by using the CLI

At the command prompt, type:

```
set service <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -downStateFlush enabled
<!--NeedCopy-->
```

To set down state flush on the service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Down State Flush.

To set down state flush on the virtual server by using the CLI

At the command prompt, type:

```
set lb vserver <name>  -downStateFlush (ENABLED | DISABLED )
<!--NeedCopy-->
```

Example:

```
set lb vserver vsvr1 -downStateFlush enabled
<!--NeedCopy-->
```

To set down state flush on the virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Down State Flush.

Graceful shutdown of services

September 21, 2020

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services. You can later enable the service by using the “enable service `<name>` command.

To avoid disrupting established sessions, you can place a service in the Transition Out of Service (TROFS) state by doing one of the following:

- Adding a TROFS code or string to the monitor—Configure the server to send a specific code or string in response to a monitor probe.
- Explicitly disable the service and:
  - Set a delay (in seconds).
  - Enable graceful shut down.

Adding a TROFS Code or String

If you bind only one monitor to a service, and the monitor is TROFS-enabled, it can place the service in the TROFS state on the basis of the server’s response to a monitor probe. This response is compared with the value in the trofsCode parameter for an HTTP monitor or the trofsString parameter for an HTTP-ECV or TCP-ECV monitor. If the code matches, the service is placed in the TROFS state. In this state, it continues to honor the persistent connections.

If multiple monitors are bound to a service, the effective state of the service is calculated on the basis of the state of all the monitors that are bound to the service. Upon receiving a TROFS response, the state of the TROFS-enabled monitor is considered as UP for the purpose of this calculation. For more information about how a NetScaler appliance designates a service as UP, see Setting a Threshold Value for the Monitors Bound to a Service.

Important:

- You can bind multiple monitors to a service, but must not TROFS-enable more than one of them.
- You can convert a TROFS-enabled monitor to a monitor that is not TROFS-enabled, but not vice versa.

To configure a TROFS code or string in a monitor by using the command line interface

At the command prompt, type one of the following commands:

```
add lb monitor <monitor-name> HTTP -trofsCode <respcode>

add lb monitor <monitor-name> HTTP-ECV  -trofsString <resp string>

add lb monitor <monitor-name> TCP-ECV  -trofsString <resp string>
<!--NeedCopy-->
```

To modify the TROFS code or string by using the command line interface

At the command prompt, type one of the following commands:

```
set lb monitor <trofs monitorname> HTTP -trofscode <newcode>

set lb monitor <trofs monitorname> HTTP-ECV -trofsstring <new string>

set lb monitor <trofs monitorname> TCP-ECV -trofsstring <new string>
<!--NeedCopy-->
```

Note: You can use the set command only if a TROFS-enabled monitor was added earlier. You cannot use this command to set the TROFS code or string for a monitor that is not TROFS-enabled.

To configure a TROFS code or string in a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. On the Monitors pane, click Add, and do one of the following:
    - Select Type as HTTP, and specify a TROFS Code.
    - Select Type as HTTP-ECV or TCP-ECV, and specify a TROFS String.

Disabling a Service

Often, however, you cannot estimate the amount of time needed for all the connections to a service to complete the existing transactions. If a transaction is unfinished when the wait time expires, shutting down the service may result in data loss. In this case, you can specify graceful shutdown for the service, so that the service is disabled only when all the current active client connections are closed by either the server or the client. See the following table for behavior if you specify a wait time in addition to graceful shutdown.

Persistence is maintained according to the specified method even if you enable graceful shutdown. The system continues to serve all the persistent clients, including new connections from the clients, unless the service is marked DOWN during the graceful shutdown state as a result of the checks made by a monitor.

The following table describes graceful shut down options.

State	Results
Graceful shutdown is enabled and a wait time is specified.	Service is shut down after the last of the current active client connections is served, even if the wait time has not expired. The appliance checks the status of the connections once every second. If the wait time expires, any open sessions are closed.
Graceful shutdown is disabled and a wait time is specified.	Service is shut down only after the wait time expires, even if all established connections are served before expiration.
Graceful shutdown is enabled and no wait time is specified.	Service is shut down only after the last of the previously established connections is served, regardless of the time taken to serve the last connection.
Graceful shutdown is disabled and no wait time is specified.	No graceful shutdown. Service is shut down immediately after the disable option is chosen or the disable command is issued. (The default wait time is zero seconds.)

To terminate existing connections when a service or a virtual server is marked DOWN, you can use the Down State Flush option. For more information, see Enabling Cleanup of Virtual Server Connections.

To configure graceful shutdown for a service by using the command line interface

At the command prompt, type the following commands to shut down a service gracefully and verify the configuration:

```
disable service <name> [<delay>] [-graceFul (YES|NO)]

show service <name>
<!--NeedCopy-->
```

Example:

```
> disable service svc1 6000 -graceFul YES
Done
>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: GOING OUT OF SERVICE (Graceful, Out Of Service in 5998 seconds)
```

```
Last state change was at Mon Nov 15 22:44:15 2010
Time since last state change: 0 days, 00:00:01.160
...
Down state flush: ENABLED

1 bound monitor:
1) Monitor Name: tcp-default
State: UP             Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - live traffic to service.
Response Time: N/A
Done

>show service svc1
svc1 (10.102.80.41:80) - HTTP
State: OUT OF SERVICE
Last state change was at Mon Nov 15 22:44:19 2010
Time since last state change: 0 days, 00:00:03.250
Down state flush: ENABLED

1 bound monitor:
1)  Monitor Name: tcp-default
State: UNKNOWN           Weight: 1
Probes: 13898    Failed [Total: 0 Current: 0]
Last response: Probe skipped - service state OFS.
Response Time: N/A
Done
<!--NeedCopy-->
```

To configure graceful shutdown for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and from the Action list, click Disable. Enter a wait time, and select Graceful.

Direct requests to a custom web page

September 21, 2020

For SureConnect to function correctly, you must set it globally. The NetScaler provides the SureConnect option to ensure the response from an application.
For more information about the SureConnect option, see "Sure Connect."

To set SureConnect on the service by using the CLI

At the command prompt, type:

```
set service <name> -sc <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -sc ON
<!--NeedCopy-->
```

To set SureConnect on the service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Sure Connect.

Enable access to services when down

September 21, 2020

You can enable access to a service when it is disabled or in a DOWN state by configuring the NetScaler appliance to use Layer 2 mode to bridge the packets sent to the service. Normally, when requests are forwarded to services that are DOWN, the request packets are dropped. When you enable the Access Down setting, however, these request packets are sent directly to the load balanced servers.

For more information about Layer 2 and Layer 3 modes, see IP Addressing.

For the appliance to bridge packets sent to the DOWN services, enable Layer 2 mode with the access-Down parameter.

To enable access down on a service by using the CLI

At the command prompt, type:

```
set service <name> -accessDown <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -accessDown YES
<!--NeedCopy-->
```

To enable access down on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Access Down.

Enable TCP buffering of responses

September 21, 2020

The NetScaler appliance provides a TCP buffering option that buffers only responses from the load balanced server. This enables the appliance to deliver server responses to the client at the maximum speed that the client can accept them. The appliance allocates from 0 through 4095 megabytes (MB) of memory for TCP buffering, and from 4 through 20480 kilobytes (KB) of memory per connection.

Note: TCP buffering set at the service level takes precedence over the global setting.
For more information about configuring TCP buffering globally, see "TCP Buffering."

To enable TCP Buffering on a service by using the CLI

At the command prompt, type:

```
set service <name> -TCPB <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -TCPB YES
<!--NeedCopy-->
```

To enable TCP Buffering on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select TCP Buffering.

Enable compression

September 21, 2020

The NetScaler appliance provides a compression option to transparently compress HTML and text files by using a set of built-in compression policies. Compression reduces bandwidth requirements and can significantly improve server responsiveness in bandwidth-constrained setups. The compression policies are associated with services bound to the virtual server. The policies determine whether a response can be compressed and send compressible content to the appliance, which compresses it and sends it to the client.

Note: For compression to function correctly, you must enable it globally.
For more information about configuring compression globally, see Compression.

To enable compression on a service by using the CLI

At the command prompt, type:

```
set service <name> -CMP <YES | NO>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CMP YES
<!--NeedCopy-->
```

To enable compression on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Compression.

Maintain client connection for multiple client requests

September 21, 2020

You can set the client keep-alive parameter to configure an HTTP or SSL service to keep a client connection to a Web site open across multiple client requests. If client keep-alive is enabled, even when the load balanced Web server closes a connection, the NetScaler appliance keeps the connection between the client and itself open. This setting allows services to serve multiple client requests on a single client connection.

If you do not enable this setting, the client will open a new connection for every request that it sends to the Web site. The client keep-alive setting saves the packet round trip time required to establish and close connections. This setting also reduces the time to complete each transaction. Client keep-alive can be enabled only on HTTP or SSL service types.

Client keep-alive set at the service level takes precedence over the global client keep-alive setting. For more information about client keep-alive, see Client Keep-Alive.

To enable client keep-alive on a service by using the CLI

At the command prompt, type:

```
set service <name> -CKA <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CKA YES
<!--NeedCopy-->
```

To enable client keep-alive on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Client Keep-Alive.

Insert the IP address of the client in the request header

September 21, 2020

A NetScaler uses the subnet IP (SNIP) address to connect to the server. The server need not be aware of the client.

However, in some situations, the server needs to be aware of the client it has to serve. When you enable the client IP setting, the appliance inserts the client's IPv4 or IPv6 address while forwarding the requests to the server. The server inserts this client IP in the header of the responses. The server is thus aware of the client.

> **Note**: To insert multiple headers, you need to perform one of the following:
>
> - Add rewrite policies to check CLIENT.IS_SSL and insert appropriate header.
> - Bind the appropriate rewrite policy for each virtual server based on the type.

To insert client IP address in the client request by using the CLI

At the command prompt, type:

```
set service <name> -CIP <Value> <cipHeader>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -CIP enabled X-Forwarded-For
<!--NeedCopy-->
```

To insert client IP address in the client request by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and open a service.
2. In Advanced Settings, select **Traffic Settings**, and select **Client IP Address**.

Use source IP address of the client when connecting to the server

September 21, 2020

You can configure the NetScaler appliance to forward packets from the client to the server without changing the source IP address. This is useful when you cannot insert the client IP address into a header, such as when working with non-HTTP services.

For more information about configuring USIP globally, see "Enabling Use Source IP Mode."

To enable USIP mode for a service by using the CLI

At the command prompt, type:

```
set service <name> -usip (YES | NO)
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -usip YES
<!--NeedCopy-->
```

To enable USIP mode for a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and open a service.
2. In Advanced Settings, in the Service Settings section, select **Use Source IP Address**.

Configure the source port for server-side connections

September 21, 2020

When the NetScaler appliance connects to a physical server, it can use the source port from client's request, or it can use a proxy port as the source port for the connection. You can set the Use Proxy Port parameter to YES to handle situations such as the following scenario:

- The NetScaler appliance is configured with two load balancing virtual servers, LBVS1 and LBVS2.
- Both the virtual servers are bound to the same service, S-ANY.
- Use (the client's) source IP address (USIP) is enabled on the service.
- Client C1 sends two requests, Req1 and Req2, for the same service.
- Req1 is received by LBVS1 and Req2 is received by LBVS2.
- LBVS1 and LBVS2 forward the request to S-ANY, and when S-ANY sends the response, they forward the response to the client.
- Consider two cases:
  - Use the client port. When the appliance uses the client port, both the virtual servers use the client's IP address (because USIP is ON) and the client's port when connecting to the server. Therefore, when the service sends the response, the appliance cannot determine which virtual server should receive the response.
  - Use proxy port. When the appliance uses a proxy port, the virtual servers use the client's IP address (because USIP is ON), but different ports when connecting to the server. Therefore, when the service sends the response, the port number identifies the virtual server that should receive the response.

However, if you require a fully transparent configuration, such as a fully transparent cache redirection configuration, you must disable the Use Proxy port Setting so that the NetScaler appliance can use the source port from the client's request.

The Use Proxy Port option becomes relevant if the use source IP (USIP) option is enabled. For TCP-based service types, such as TCP, HTTP, and SSL, the option is enabled by default. For UDP-based service types, such as UDP and DNS, including ANY, the option is disabled by default. For more information about the USIP option, see "Enabling Use Source IP Mode."

You can configure the Use Proxy Port setting either globally or on a given service.

Configure the use proxy port setting on a service

You configure the Use ProxyPort settingon the service if you want to override the global setting.

To configure the Use Proxy Port setting on a service by using the CLI

At the command prompt, type:

```
set service <name> -useProxyPort (YES | NO)
<!--NeedCopy-->
```

Example:

```
set service svc1 -useproxyport YES
Done

show service svc1
svc1 (10.102.29.30:80) - HTTP
State: UP
. . .
Use Source IP: YES Use Proxy Port: YES
. . .
Done
<!--NeedCopy-->
```

To configure the Use Proxy Port setting on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Proxy Port.

Configure the use proxy port setting globally

You configure the Use Proxy Port setting globally if you want to apply the setting to all the services on the NetScaler appliance. The global setting is overridden by service-specific Use Proxy Port settings.

To configure the Use Proxy Port setting globally by using the CLI

At the command prompt, type the following commands to configure the Use Proxy Port setting globally and verify the configuration:

set ns param -useproxyport (ENABLED	DISABLED)

-
- show ns param

Example:

```
set ns param -useproxyport ENABLED

Done

show ns param
Global configuration settings:
```

```
. . .
Use Proxy Port: ENABLED
Done
<!--NeedCopy-->
```

To configure the Use Proxy Port setting globally by using the GUI

Navigate to System > Settings > Change global system settings, and select or clear Use Proxy Port.

Set a limit on the number of client connections

September 21, 2020

You can specify a maximum number of client connections that each load balanced server can handle. The NetScaler appliance then opens client connections to a server only until this limit is reached. When the load balanced server reaches its limit, monitor probes are skipped, and the server is not used for load balancing until it has finished processing existing connections and frees up capacity.

For more information on the Maximum Client setting, see Load Balancing Domain-Name Based Services.

Note: Connections that are in the process of closing are not considered for this limit.

To set a limit to the number of client connections by using the CLI

At the command prompt, type:

```
set service <name> -maxclient <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -maxClient 1000
<!--NeedCopy-->
```

To set a limit to the number of client connections by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Clients.

Set a limit on number of requests per connection to the server

September 21, 2020

The NetScaler appliance can be configured to reuse connections to improve performance. In some scenarios, however, load balanced Web servers may have issues when connections are reused for too many requests. For HTTP or SSL services, use the max request option to limit the number of requests sent through a single connection to a load balanced Web server.

Note: You can configure the max request option for HTTP or SSL services only.

To limit the number of client requests per connection by using the CLI

At the command prompt, type:

```
set service <ServiceName> -maxReq <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -maxReq 100
<!--NeedCopy-->
```

To limit the number of client requests per connection by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Requests.

Set a threshold value for the monitors bound to a service

September 21, 2020

The NetScaler appliance designates a service as UP only when the sum of the weights of all monitors bound to it and that are UP is equal to or greater than the threshold value configured on the service. The weight for a monitor specifies how much that monitor contributes to designating the service to which it is bound as UP.

By default, the monitor threshold is set to 0 and monitor weights are set to 1.
All monitors have equal weightage then and a service can go DOWN when any one of the monitors goes DOWN.

For example, assume that three monitors, named Monitor-HTTP-1, Monitor-HTTP-2, and Monitor-HTTP-3 respectively, are bound to Service-HTTP-1, and that the threshold configured on the service is three. Suppose the following weights are assigned to each monitor:

- The weight of Monitor-HTTP-1 is 1.
- The weight of Monitor-HTTP-2 is 3.
- The weight of Monitor-HTTP-3 is 1.

The service is marked UP only when one of the following is true:

- Monitor-HTTP-2 is UP.
- Monitor-HTTP-2 and Monitor-HTTP-1 or Monitor-HTTP-3 are UP
- All three monitors are UP.

To set the monitor threshold value on a service by using the CLI

At the command prompt, type:

```
set service <name> -monThreshold <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -monThreshold 100
<!--NeedCopy-->
```

To set the monitor threshold value on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Monitor Threshold.

Set a timeout value for idle client connections

September 21, 2020

You can configure the service with a time-out value to terminate any idle client connections when the configured time elapses. If the client is idle during the configured time, the NetScaler appliance closes the client connection.

To set a timeout value for idle client connections by using the CLI

At the command prompt, type:

```
set service <name> -cltTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -cltTimeout 100
<!--NeedCopy-->
```

To set a timeout value for idle client connections by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Client Idle Time-out.

Set a timeout value for idle server connections

September 21, 2020

You can configure a service with a timeout value to terminate any idle server connections when the configured time (in seconds) elapses. If the server is idle for the configured amount of time, the NetScaler appliance closes the server connection.

To set a timeout value for idle server connections by using the CLI

At the command prompt, type:

```
set service <name> -svrTimeout <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -svrTimeout 100
<!--NeedCopy-->
```

To set a timeout value for idle server connections by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Server Idle Time-out.

Set a limit on the bandwidth usage by clients

September 21, 2020

In some cases, servers may have limited bandwidth to handle client requests and may become overloaded. To prevent overloading a server, you can specify a maximum limit on the bandwidth, in Kbps, processed by the server. The NetScaler appliance forwards requests to a load balanced server only until this limit is reached.

To set a maximum bandwidth limit on a service by using the CLI

At the command prompt, type:

```
set service <name> -maxBandwidth <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -maxBandwidth 100
<!--NeedCopy-->
```

To set a maximum bandwidth limit on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Bandwidth.

Redirect client requests to a cache

September 21, 2020

You can configure a service to redirect client requests to a cache, and forward the non-cacheable requests to a service chosen by the configured load balancing method.

To set cache redirection on a service by using the CLI

At the command prompt, type:

```
set service <name> -cacheable <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-HTTP-1 -cacheable YES
<!--NeedCopy-->
```

To set cache redirection on a service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and set the Cache Type.

Retain the VLAN identifier for VLAN transparency

September 21, 2020

You can configure a load balancing virtual server to retain the client's VLAN identifier in packets that are to be forwarded to servers. The virtual server must be a wildcard virtual server of type ANY, and must be functioning in MAC mode.

To configure a load balancing virtual server to retain the client VLAN ID by using the CLI

At the command prompt, type the following command to configure a load balancing virtual server to retain the client VLAN ID and verify the configuration:

```
set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED

show lb vserver <name>
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure a load balancing virtual server to retain the client VLAN ID by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and select Retain VLAN ID.

Configure automatic state transition based on percentage health of bound services

September 21, 2020

You can configure a load balancing virtual server to automatically transition from the UP state to the DOWN state if the percentage of active services falls below a configured threshold. For example, if you bind 10 services to a load balancing virtual server and configure a threshold of 50% for that virtual server, it transitions from UP to DOWN if six or more services are DOWN. When the percentage health rises above the threshold value, the virtual server returns to the UP state.

You can also enable an SNMP alarm called ENTITY-STATE if you want the NetScaler appliance to notify you when the percentage health of bound services causes a virtual server to change state.

To configure percentage based automatic state transition by using the CLI

At the command prompt, type the following commands to configure automatic state transition for a virtual server and verify the configuration:

```
set lb vserver <name> -healthThreshold <positive_integer>

show lb vserver <name>
<!--NeedCopy-->
```

To configure percentage based automatic state transition by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In Advanced Settings, select Traffic Settings, and set a Health Threshold.

To enable the ENTITY-STATE alarm by using the CLI

At the command prompt, type the following commands to enable the ENTITY-STATE SNMP alarm and verify the configuration:

```
enable snmp alarm ENTITY-STATE

show snmp alarm
<!--NeedCopy-->
```

To enable the ENTITY-STATE alarm by using the GUI

1. Navigate to System > SNMP > Alarms.
2. Select ENTITY-STATE and, in the Action list, select Enable.

Built-in monitors

September 21, 2020

The NetScaler appliance contains a number of built-in monitors that you can use to monitor your services. These built-in monitors handle most of the common protocols. You cannot modify or remove the built-in monitors; you can only bind a built-in monitor to a service and unbind it from the service.

Note: You can create a custom monitor based on a built-in monitor. To learn how to create custom monitors, see
Configuring Monitors in a Load Balancing Setup.

TCP-based application monitoring

September 21, 2020

The NetScaler appliance has two built-in monitors that monitor TCP-based applications: tcp-default and ping-default. When you create a service, the appropriate default monitor is bound to it automatically, so that the service can be used immediately if it is UP. The tcp-default monitor is bound to all TCP services; the ping-default monitor is bound to all non-TCP services.

You cannot delete or modify default monitors. When you bind any other monitor to a TCP service, the default monitor is unbound from the service. The following table lists the monitor types, and the parameters and monitoring processes associated with each type.

Monitor type	Specific parameters	Process
tcp	Not applicable	The NetScaler appliance establishes a 3-way handshake with the monitor destination, and then closes the connection. If the appliance observes TCP traffic to the destination, it does not send TCP monitoring requests. This occurs if LRTM is disabled. By default, LRTM is disabled on this monitor.
http	httprequest [“HEAD /”] - HTTP request that is sent to the service. respcode [200] - A set of HTTP response codes are expected from the service.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.
tcp-ecv	send [””] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes. recv [””] - expected response from the service. The maximum permissible length of the string is 128 K bytes.	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send specific data to the service and expects a specific response through the receive parameter. Different servers send different sizes of segments. However, the pattern must be within 16 TCP segments.

Monitor type	Specific parameters	Process
http-ecv	send [""] - HTTP data that is sent to the service; recv [""] - the expected HTTP response data from the service	The NetScaler appliance establishes a 3-way handshake with the monitor destination. When the connection is established, the appliance uses the send parameter to send the HTTP data to the service and expects the HTTP response that the receive parameter specifies. (HTTP body part without including HTTP headers). Empty response data matches any response. Expected data may be anywhere in the first 24K bytes of the HTTP body of the response.
ping	Not Applicable	The NetScaler appliance sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.

To configure built-in monitors for TCP-based applications, see Configuring Monitors in a Load Balancing Setup.

SSL service monitoring

September 21, 2020

The NetScaler appliance has built-in secure monitors, TCPS and HTTPS. You can use the secure monitors to monitor HTTP as well as non-HTTP traffic. To configure a secure HTTP monitor, select the monitor type as HTTP, and then set the secure flag. To configure a secure TCP monitor, select the monitor type as TCP, and then set the secure flag.The secure monitors work as described below:

- **Secure TCP monitoring.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. After the handshake is over, the appliance closes the connection.
- **Secure HTTP monitoring.** The NetScaler appliance establishes a TCP connection. After the connection is established, the appliance performs an SSL handshake with the server. When the SSL connection is established, the appliance sends HTTP requests over the encrypted channel and checks the response codes.

The following table describes the available built-in monitors for monitoring SSL services.

Monitor type	Probe	Success criteria (Direct condition)
TCP	TCP connection; SSL handshake	Successful TCP connection established and successful SSL handshake.
HTTP	TCP connection; SSL handshake; Encrypted HTTP request	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP response code in server HTTP response is encrypted.
TCP-ECV	TCP connection; SSL handshake (Data sent to a server is encrypted.)	Successful TCP connection is established, successful SSL handshake is performed, and expected TCP data is received from the server.
HTTP-ECV	TCP connection; SSL handshake (Encrypted HTTP request)	Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP data is received from the server.

Sample configuration for HTTPS-ECV health check monitor

HTTP services have predefined monitors capable of Extended Content Verification (ECV).
These monitors are used when a validation is required beyond a successful TCP connection. These monitors validate the service as UP, when all the following criteria are met:

- A successful TCP connection.

- A particular type of request must be generated.
- A specific message is expected in reply from the **Receive String**.

For these monitors, a request string is configured along with a reply string. If the reply string received by the NetScaler monitor matches the configured string, then the service is marked UP.

Bind a monitor to a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, create a service, and specify the protocol as **SSL**. Click **OK**.
2. Click in the **Service to Load Balancing Monitor Binding** pane, and click **Add Binding**.
3. Choose the monitor type as **HTTPS-ECV** and click **Edit**.
4. In the **Configure Monitor** pane under **Basic Parameters** tab, enter values for the following parameters:
    - **Send String** – The string that the monitor must send to the service.
    - **Receive String** – The string that the monitor must receive to mark the service as UP.
5. Click **OK** to complete the monitor configuration.
6. Click **Select**.
7. Click **Bind** to bind the **HTTPS-ECV** monitor to the service.
8. Click **Close**.

> **Note**
>
> Prior to NetScaler release 12.0 build 56.20, Basic Parameters and Advanced Parameters are named Standard Parameters and Special Parameters respectively.

Bind a monitor to a service by using the CLI

At the command prompt, type:

```
bind service <servicename> -monitorName https-ecv
<!--NeedCopy-->
```

Example:

```
bind services1 -monitorName https-ecv
<!--NeedCopy-->
```

FTP service monitoring

September 21, 2020

To monitor FTP services, the NetScaler appliance opens two connections to the FTP server. It first connects to the control port, which is used to transfer commands between a client and an FTP server. After it receives the expected response, it connects to the data port, which is used to transfer files between a client and an FTP server. Only when the FTP server responds as expected, on both the connections, it is marked UP.

Note: Monitor probes originate from the NSIP address.

The NetScaler appliance has two built-in monitors for FTP services: the FTP monitor and the FTP-EXTENDED monitor. The FTP-EXTENDED monitor is a scriptable monitor. It uses the nsftp.pl script. The FTP-EXTENDED monitor script is enhanced to send secure probes to FTP services. You can create a monitor of type FTP-EXTENDED. The nsftp.pl script is automatically taken from the default directory.

To send secure FTP probes to FTP services by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <type> -username <string> -password <string> -filename <filename>
<!--NeedCopy-->
```

Example

```
add monitor mon1 FTP-EXTENDED -username root -password freebsd -filename fsdf
<!--NeedCopy-->
```

To send secure FTP probes to FTP services by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Specify the monitor type as **FTP-EXTENDED**, and set the parameters.
3. In **Special Parameters**, specify a **File Name**, **User Name**, and **Password**.

To configure built-in monitors to check the state of FTP services, see Configuring Monitors in a Load Balancing Setup.

Secure monitoring of servers by using SFTP

September 21, 2020

A user script ‘nssftp.pl’ is added to support SSH File Transfer Protocol (SFTP) monitoring. It is available in the current list of in-built NetScaler user monitors and is located in the /netscaler/monitors directory. The SFTP monitor uses the specified username and password to check if the file is present on the server.

To configure secure monitoring using SFTP by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <type> -scriptName <string> -scriptArgs <string> -secure ( YES | NO )
<!--NeedCopy-->
```

Example:

```
add monitor SFTP_MON USER –scriptname nssftp.pl –scriptargs “file=example.txt;user=sam;password=sam_passwd”
<!--NeedCopy-->
```

To configure secure monitoring using SFTP by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors** and in **Type** specify **USER**.
2. In **Special Parameters**, in **Script Name**, select nssftp.pl.
3. Specify the **Script Arguments**.

Set SSL parameters on a secure monitor

September 21, 2020

> **Important**
>
> This feature is supported only on the new Default profiles. For more information about these profiles, see Enhanced SSL Profiles Infrastructure Overview.

A monitor inherits either the global settings or the settings of the service to which it is bound. If a monitor is bound to a non-SSL or non-SSL_TCP service, such as SSL_BRIDGE, you cannot configure it with SSL settings such as the protocol version or the ciphers to be used. Therefore, if your deployment requires SSL-based monitoring of the back-end servers, the monitoring is ineffective.

You can have more control over SSL-based monitoring of back-end servers, by binding an SSL profile to a monitor. An SSL profile contains SSL parameters, cipher bindings, and ECC bindings. For example, you can set server authentication, ciphers, and protocol version in an SSL profile and bind the profile to a monitor. Note that to perform server authentication, you must also bind a CA certificate to a monitor. To perform client authentication, you must bind a client certificate to the monitor. New parameters for the “bind lb monitor” command enable you to do so.

> **Note:**
> The SSL settings take effect only if you add a secure monitor. Also, the SSL profile type must be **BackEnd**.

Monitor Types that Support SSL Profiles

SSL profiles can be bound to the following monitor types:

- HTTP
- HTTP-ECV
- TCP
- TCP-ECV
- HTTP-INLINE

To specify an SSL profile while adding a monitor by using the command line

At the command prompt, type:

```
add lb monitor <monitorName> <type> -secure  YES   -sslprofile <string>
set lb monitor <monitorName> <type> -secure YES -sslprofile <string>
<!--NeedCopy-->
```

Example:

```
add ssl profile prof1 -sslProfileType  BackEnd
add lb monitor mon1 HTTP -secure  YES -sslprofile prof1
<!--NeedCopy-->
```

To bind a certificate-key pair to a monitor by using the command line

At the command prompt, type:

```
bind monitor  <monitor name> -certkeyName <string>  [(-CA  [-crlCheck (
    Mandatory | Optional ) | -ocspCheck ( Mandatory | Optional )]
<!--NeedCopy-->
```

SIP service monitoring

September 21, 2020

A NetScaler has two built-in monitors that you can use to monitor SIP services: the **SIP-UDP** and **SIP-TCP** monitors. A SIP monitor periodically checks the SIP service to which the SIP monitor is bound, by sending SIP request methods to the SIP service. If the SIP service replies with a response code, the monitor marks the service as UP. If the SIP service does not respond, or responds incorrectly, it is marked as DOWN.

Parameter	Specifies
sipURI	SIP addressing schema of the SIP server.
sipmethod	Type of SIP request used to probe the SIP service. Specify one of the following methods: INVITE, OPTION (the default), REGISTER
respcode	SIP response code with which the SIP service responds the probe request. Default: 200.

RADIUS service monitoring

September 21, 2020

The NetScaler appliance RADIUS monitor periodically checks the state of the RADIUS service to which it is bound by sending an authentication request to the service. The RADIUS server authenticates the RADIUS monitor and sends a response. By default, the monitor expects to receive a response code of 2, the default Access-Accept response, from the RADIUS server. As long as the monitor receives the appropriate response, it marks the service UP.

Note: RADIUS monitor supports only PAP type authentication.

- If the client authenticated successfully, the RADIUS server sends an Access-Accept response. The default access-accept response code is 2, and this is the code that the appliance uses.
- If the client fails to authenticate successfully (such as when there is a mismatch in the user name, password, or secret key), the RADIUS server sends an Access-Reject response. The default access-reject response code is 3, and this is the code that the appliance uses.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
radKey	Shared secret key value that the RADIUS server uses during client authentication.
radNASid	NAS-ID that is encapsulated in the payload when an access request is made.
radNASip	The IP address that is encapsulated in the payload when an access-request is made. When radNASip is not configured, the NetScaler appliance sends the mapped IP address (MIP) to the RADIUS server as the NAS IP address.

To monitor a RADIUS service, you must configure the RADIUS server to which it is bound as follows:

1. Add the user name and password of the client that the monitor will use for authentication to the RADIUS authentication database.
2. Add the IP address and secret key of the client to the appropriate RADIUS database.
3. Add the IP addresses that the appliance uses to send RADIUS packets to the RADIUS database. If the NetScaler appliance has more than one mapped IP address, or if a subnet IP address (SNIP) is used, you must add the same secret key for all of the IP addresses.

   Caution: If the IP address used by the appliance are not added to the RADIUS database, the RADIUS server will discard all packets.

To configure built-in monitors to check the state of RADIUS server, see Configuring Monitors in a Load Balancing Setup.

Monitor accounting information delivery from a RADIUS server

September 21, 2020

You can configure a monitor called a *RADIUS accounting* monitor to determine whether the Radius server used for Authentication, Authorization, and Accounting (AAA) is delivering accounting information as expected. The monitor is of type RADIUS_ACCOUNTING. The probe is generated by a Perl script called nsbmradius.pl, which is located in the /nsconfig/monitors/ directory. The script sends successive accounting request probes to the RADIUS server. The probe is considered successful only if the RADIUS accounting server responds with a packet whose Code field is set to 5, which, according to RFC 2866, indicates an Accounting-Response packet.

When configuring a RADIUS accounting monitor, you must specify a secret key. You can specify optional parameters, each of which represents a RADIUS attribute, such as Acct-Status-Type and Framed-IP-Address. For information about these attributes, see RFC 2865, "Remote Authentication Dial In User Service (RADIUS)," and RFC 2866, "RADIUS Accounting."

To configure a RADIUS accounting monitor by using the CLI

At the command prompt, type the following commands to configure a RADIUS accounting monitor and verify the configuration:

```
add lb monitor <monitorName> RADIUS_ACCOUNTING [-userName <string>] {
 -password  }
  {
 -radKey  }
  [-radNASip <ip_addr>] [-radAccountType <positive_integer>] [-
     radFramedIP <ip_addr>] [-radAPN <string>] [-radMSISDN <string>] [-
     radAccountSession <string>]

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example

```
add lb monitor radAccntMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-
   zW13sM"
<!--NeedCopy-->
```

DNS and DNS-TCP service monitoring

September 21, 2020

The NetScaler appliance has two built-in monitors that can be used to monitor DNS services: DNS and DNS-TCP. When bound to a service, either monitor periodically checks the state of that DNS service by sending a DNS query to it. The query resolves to an IPv4 or IPv6 address. That IP address is then checked against the list of test IP addresses that you configure. The list can contain up to five IP addresses. If the resolved IP address matches at least one IP address on the list, the DNS service is marked as up. If the resolved IP does not match any IP addresses on the list, the DNS service is marked as down.

Parameter	Description
query	The DNS query (domain name) sent to the DNS service that is being monitored. Default value: "\007" If the DNS query succeeds, the service is marked as UP; otherwise, it is marked as DOWN. For a reverse monitor, if the DNS query succeeds, the service is marked as DOWN; otherwise, it is marked as UP. If no response is received, the service is marked as DOWN.
queryType	The type of DNS query that is sent. Possible values: Address, Zone.
IPAddress	List of IP addresses that are checked against the response to the DNS monitoring probe.
IPv6	Select this check box if the IP address uses IPv6 format.

To configure the built-in DNS or DNS-TCP monitors, see Configuring Monitors in a Load Balancing Setup.

LDAP service monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor LDAP services: the LDAP monitor. It periodically checks the LDAP service to which it is bound by authenticating and sending a search query to it. If the search is successful, the service is marked UP. If the LDAP server does not locate the entry, a failure message is sent to the LDAP monitor, and the service is marked DOWN.

You configure the LDAP monitor to define the search that it should perform when sending a query. You can use the Base DN parameter to specify a location in the directory hierarchy where the LDAP server should start the test query. You can use the Attribute parameter to specify an attribute of the target entity.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
baseDN	Base name for the LDAP monitor from where the LDAP search must start. If the LDAP server is running locally, the default value of base is dc=netscaler, dc=com.
bindDN	BDN name for the LDAP monitor.
filter	Filter for the LDAP monitor.
password	Password used in monitoring LDAP servers.
attribute	Attribute for the LDAP monitor.

To configure the built-in LDAP monitor, see Configuring Monitors in a Load Balancing Setup.

MySQL service monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor MySQL services: the MySQL monitor. It periodically checks the MySQL service to which it is bound by sending a search query to it. If the search is successful, the service is marked UP. If the MySQL server does not respond or the search fails, a failure message is sent to the MySQL monitor, and the service is marked DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
database	Database that is used for the MySQL monitor.
sqlQuery	SQL query that is used for the MySQL monitor.

To configure built-in MySQL monitor, see Configuring Monitors in a Load Balancing Setup.

SNMP service monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor SMNP services: the SNMP monitor. It periodically checks the SNMP agent on the service to which it is bound by sending a query for the enterprise identification ID (OID) that you configure for monitoring. If the query is successful, the service is marked UP. If the SNMP service finds the OID that you specified, the query succeeds and the SNMP monitor marks the service UP. If it does not find the OID, the query fails and the SNMP monitor marks service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
SNMPOID	OID that is used for the SNMP monitor.
snmpCommunity	Community that is used for the SNMP monitor.
snmpThreshold	Threshold that is used for the SNMP monitor.
snmpVersion	SNMP version that is used for load monitoring. Possible Values: V1, V2.

To configure the built-in SNMP monitor, see Configuring Monitors in a Load Balancing Setup.

NNTP service monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor NNTP services: the NNTP monitor. It periodically checks the NNTP service to which it is bound by connecting to the service and checking for the existence of the newsgroup that you specify. If the newsgroup exists, the search is successful and the service is marked UP. If the NNTP service does not respond or the search fails, the service is marked DOWN.

Note: Monitor probes originate from the NSIP address.

The NNTP monitor can optionally be configured to post a test message to the newsgroup as well.

Parameter	Specifies
userName	User name on the RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3 server. This user name is used in the probe.
password	Password used in monitoring RADIUS/NNTP/FTP/FTP-EXTENDED/MYSQL/POP3/LDAP servers.
group	Group name to be queried for NNTP monitor.

To configure the built-in NNTP monitor, see Configuring Monitors in a Load Balancing Setup.

POP3 service monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor POP3 services: the POP3 monitor. It periodically checks the POP3 service to which it is bound by opening a connection with a POP3 server. If the POP3 server responds with the correct response codes within the configured time period, it marks the service UP. If the POP3 service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
userName	User name POP3 server. This user name is used in the probe.
password	Password used in monitoring POP3 servers.
scriptName	The path and name of the script to execute.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in POP3 monitor, see Configuring Monitors in a Load Balancing Setup.

SMTP service monitoring

June 15, 2021

The NetScaler appliance has a built-in monitor that can be used to monitor SMTP services: the SMTP monitor. It periodically checks the SMTP service to which it is bound by opening a connection with it and conducting a series of handshakes to ensure that the server is operating correctly. If the SMTP service completes the handshakes properly, the monitor marks the service UP. Else, if the SMTP service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NSIP address.

Parameter	Specifies
scriptName	The path and name of the script to run.
dispatcherIP	The IP Address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.

To configure the built-in SMTP monitor, see Configuring Monitors in a Load Balancing Setup.

RTSP service monitoring

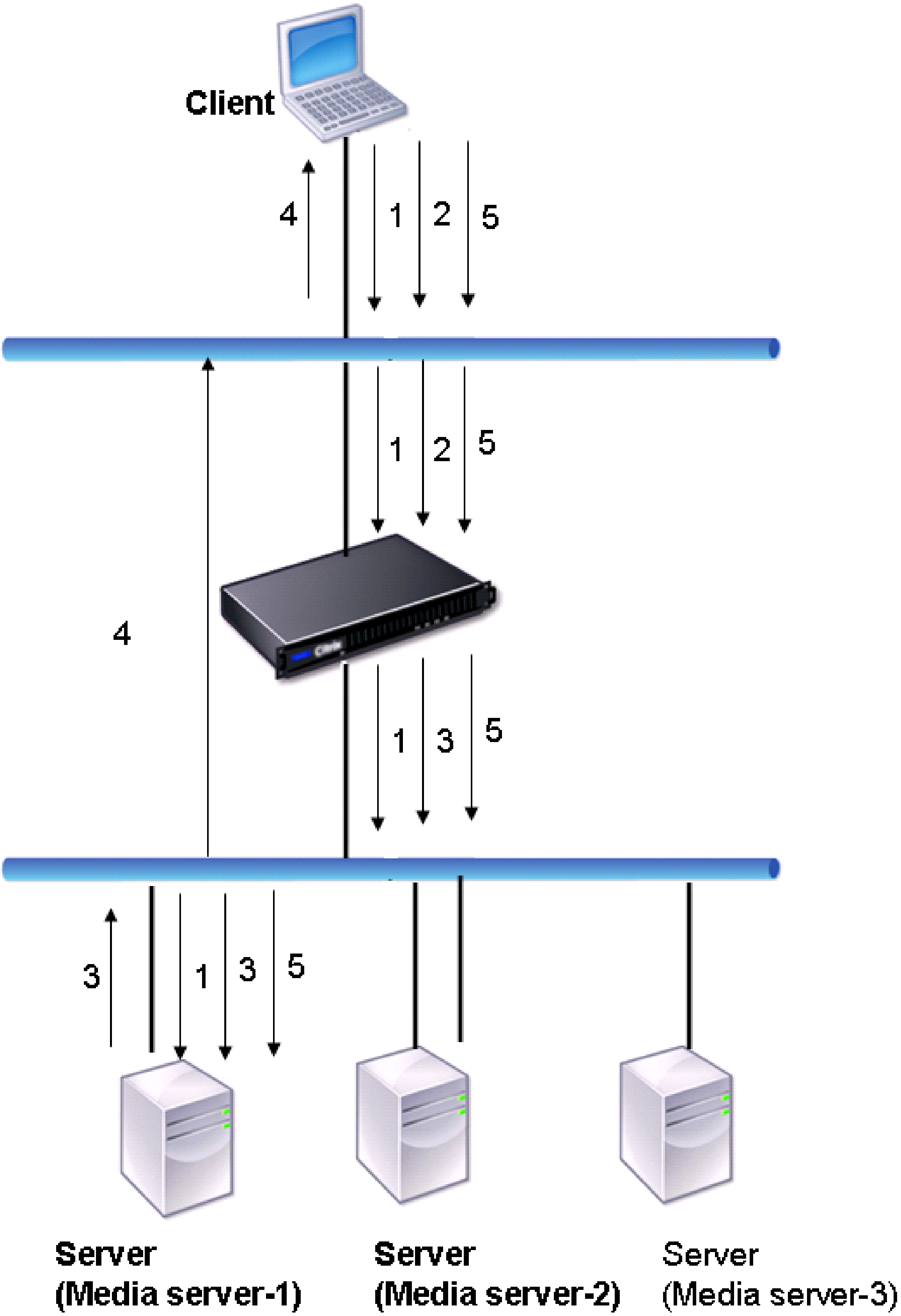
September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor RTSP services: the RTSP monitor. It periodically checks the RTSP service to which it is bound by opening a connection with the load balanced RTSP server. The type of connection that it opens, and the response that it expects, differs depending upon the network configuration. If the RTSP service responds as expected within the configured time period, it marks the service UP. If the service does not respond, or responds incorrectly, it marks the service DOWN.

The NetScaler appliance can be configured to load balance RTSP servers using two topologies: NAT-off and NAT-on. RTSP servers send their responses directly to the client, bypassing the appliance. The appliance must be configured to monitor RTSP services differently depending upon which topology your network uses. The appliance can be deployed either in inline or non-inline mode in both NAT-off and NAT-on mode.

In NAT-off mode, the appliance operates as a router: it receives RTSP requests from the client and routes them to the service that it selects using the configured load balancing method. If your load balanced RTSP servers are assigned publicly accessible FQDNs in DNS, the load balanced servers send their responses directly to the client, bypassing the appliance. The following figure demonstrates this configuration.

Figure 1. RTSP in NAT-off Mode


Client
Server
(Media server-1)
Server
(Media server-2)
Server
(Media server-3)

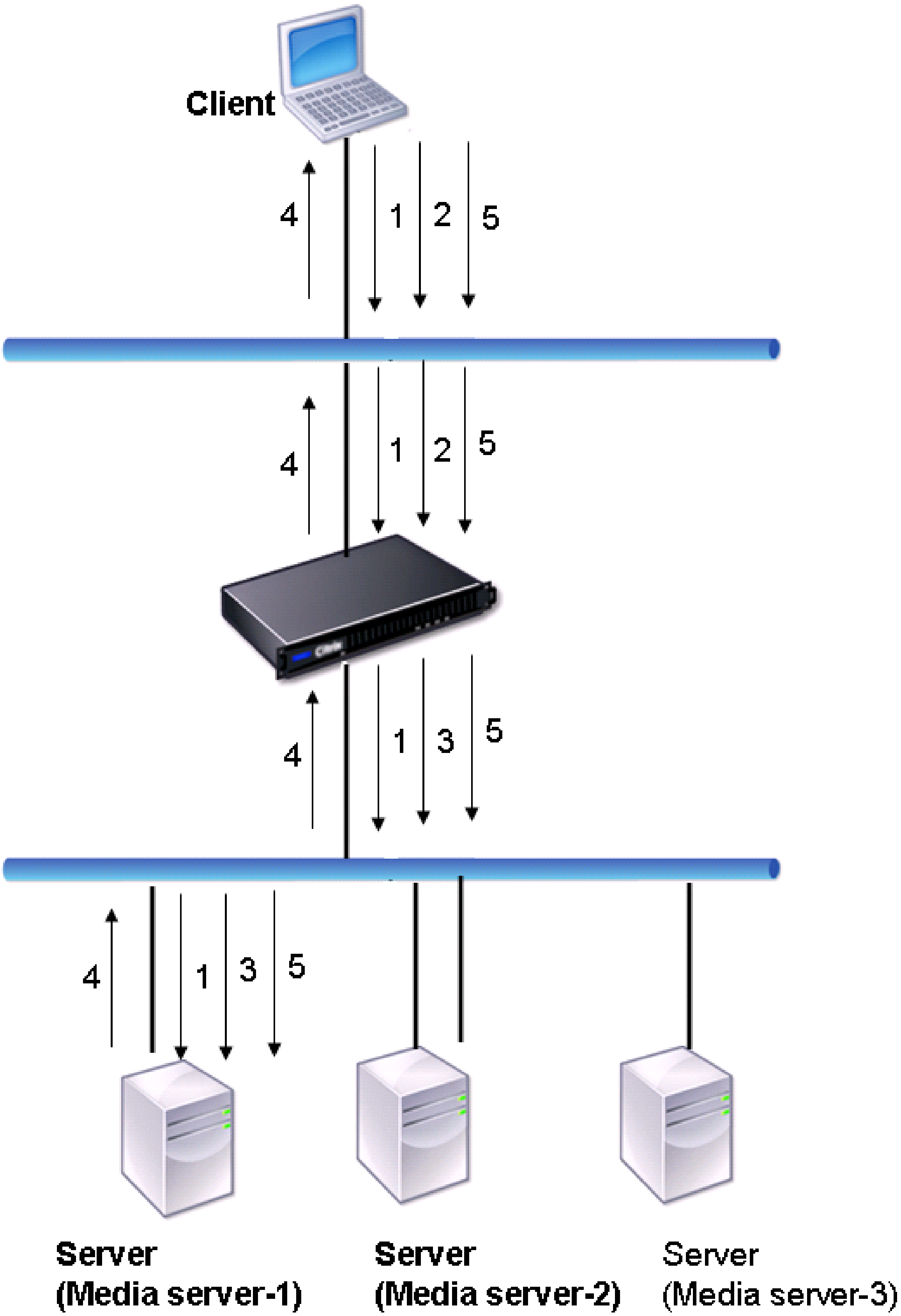
The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:
   - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
   - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.

   Note: The appliance does not perform NAT to identify the RTSP connection, because the RTSP connections bypass it.
4. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the media server. Media Server-1 performs the requested actions, such as play, forward, or rewind.

In NAT-on mode, the appliance receives RTSP requests from the client and routes those requests to the appropriate media server using the configured load balancing method. The media server then sends its responses to the client through the appliance, as illustrated in the following diagram.

Figure 2. RTSP in NAT-on Mode


Client
Server
(Media server-1)
Server
(Media server-2)
Server
(Media server-3)

The flow of requests and responses in this scenario is as follows:

1. The client sends a DESCRIBE request to the appliance. The appliance uses the configured load balancing method to choose a service, and routes the request to Media Server-1.
2. The client sends a SETUP request to the appliance. If the RTSP session ID is exchanged in the DESCRIBE request, the appliance, using the RTSPSID persistence, routes the request to Media Server-1. If the RTSP session ID is exchanged in the SETUP request, the appliance does one of the following:
    - If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
    - If the request arrives on a different TCP connection, it uses the configured load balancing method to choose a service, and sends the request to that service, not maintaining persistence. This means that the request may be sent to a different service.
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.
4. The appliance performs NAT to identify the client for RTSP data connections, and the RTSP connections pass through the appliance and are routed to the correct client.
5. For subsequent requests, the client then uses the session ID to identify the session and send control messages to the appliance. The appliance uses RTSPSID persistence to identify the appropriate service, and routes the request to Media Server-1. Media Server-1 performs the requested action, such as play, forward, or rewind.

The RTSP monitor uses the RTSP protocol to evaluate the state of the RTSP services. The RTSP monitor connects to the RTSP server and conducts a sequence of handshakes to ensure that the server is operating correctly.

Parameter	Specifies
rtspRequest	The RTSP request string that is sent to the RTSP server (for example, OPTIONS *). The default value is 07. The length of the request must not exceed 163 characters.
respCode	Set of response codes that are expected from the service.

For instructions on configuring an RTSP monitor, see Configuring Monitors in a Load Balancing Setup.

XML broker service monitoring

September 21, 2020

The NetScaler appliance has a built-in monitor type, CITRIX-XML-SERVICE, with which you can create monitors to monitor the XML Broker services. The XML Broker services are used by Citrix XenApp. The monitor opens a connection to the service and periodically probes the XML services to which it is bound. If the server responds as expected within the configured time period, the monitor marks the service UP. If the service does not respond, or responds incorrectly, the monitor marks the service DOWN.

To configure a CITRIX-XML-SERVICE monitor, you need to specify the application name in addition to setting the standard parameters. The application name is the name of the application that has to be run to monitor the state of the XML Broker service. The default application is Notepad.

To configure monitors for XML Broker services, see "Configuring Monitors in a Load Balancing Setup."

> **Note**
>
> The parameter "Application Name" for Citrix-XML-Service monitor is invalid for XA and XD versions 7 and later. It is recommended not to use this parameter in XA/XD 7. In case you configure this parameter, this parameter is not used internally. The probing criteria is different starting XA/XD 7. However, you can use the "Application Name" parameters in versions earlier to XA/XD 7.

ARP request monitoring

September 21, 2020

The NetScaler appliance has one built-in monitor that can be used to monitor ARP requests: the ARP monitor. This monitor periodically sends an ARP request to the service to which it is bound, and listens for the expected response. If it receives the expected response, it marks the service UP. If it receives no response or the wrong response, it marks the service DOWN.

ARP locates a hardware address for a load balanced server when only the network layer address is known. ARP works with IPv4 to translate IP addresses to Ethernet MAC addresses. ARP monitoring is not relevant to IPv6 networks, and is therefore not supported on those networks.

There are no special parameters for the ARP monitor.

For instructions on configuring an ARP monitor, see Configuring Monitors in a Load Balancing Setup.

XenDesktop Delivery Controller service monitoring

September 21, 2020

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and the XenDesktop Delivery Controller servers deployed by Citrix XenDesktop environment. The NetScaler appliance provides a built-in monitor, CITRIX-XD-DDC monitor, which monitors the XenDesktop Delivery Controller servers. In addition to the health check, you can also verify whether the probe is sent by a valid user of the XenDesktop Delivery Controller server.

The monitor sends a probe to the XenDesktop Delivery Controller server in the form of an XML message. If the server responds to the probe with the identity of the server farm, the probe is considered to be successful and the server's status is marked as UP. If the HTTP response does not have a success code or the identity of the server farm is not present in the response, the probe is considered to be a failure and the server's status is marked as DOWN.

The Validate Credentials option determines the probe to be sent by the monitor to the XenDesktop Delivery Controller server, that is, whether to request only the server name or to also validate the login credentials.

Note: Regardless of whether or not the user credentials (user name, password and domain) are specified on the CITRIX-XD-DDC monitor, the XenDesktop Delivery Controller server validates the user credentials only if the option to validate credentials is enabled on the monitor.

If you use the wizard for configuring the load balancing of the XenDesktop servers, the CITRIX-XD-DDC monitor is automatically created and bound to the XenDesktop Delivery Controller services.

To add an XD-DDC monitor with the validate credentials option by using the command line interface

At the command prompt, type the following commands to add an XD-DDC monitor and verify the configuration:

```
add lb monitor <monitorName> <monitorType> -userName <userName> -password <password> -ddcDomain <ddc_domain_name> -validateCred YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
 > add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -password E12Dc35450a1 -ddcDomain dhop -validateCred YES
 Done
> show lb monitor xdddcmon
1) Name......:xdddcmon Type......:CITRIX-XD-DDC State......: ENABLED

Standard parameters:
Interval......:..5 sec...Retries......:..3
Response timeout......:..2 sec...Down time......:..30 sec
Reverse......:..NO...Transparent......:..NO
Secure......:..NO...LRTM......:..ENABLED
Action......:..Not applicable...Deviation......:..0 sec
Destination IP......:..Bound service
Destination port......:..Bound service
Iptunnel......:..NO
TOS......:..NO...TOS ID......:..0
SNMP Alert Retries......:..0...Success Retries......:..1
Failure Retries......:..0

Special parameters:
User Name..........:"Administrator"
Password...........:*****
DDC Domain.........: "dhop"
Done
<!--NeedCopy-->
```

To specify the validate credentials option on an XD-DDC monitor by using the command line interface

At the command prompt, type:

```
set lb monitor <monitorName> <monitorType> -userName -password -
   ddcDomain <ddc_domain_name> -validateCred YES
<!--NeedCopy-->
```

Example:

```
 set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName
    Administrator -password D123S1R2A123 -ddcDomain dhop -validateCred
    YES
Done
<!--NeedCopy-->
```

To configure an XD-DDC monitor with the validate credentials option by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type Citrix-XD-DDC.

Citrix StoreFront stores monitoring

September 21, 2020

You can configure a user monitor for a Citrix Storefront store. The monitor determines the state of the StoreFront store by successively probing the account service, discovery service, and authentication endpoint (if the Citrix Storefront Store is an authenticated store). If any of those services do not respond to the probe, the monitor probe fails, and the StoreFront store is marked as DOWN. The monitor sends probes to the IP address and port of the bound service. For more information, see Citrix StoreFront Store Services API.

Note: Monitor probes originate from the NSIP address. However, if the subnet of a StoreFront server is different from that of the appliance, then the subnet IP (SNIP) address is used.

Beginning with release 10.1 build 120.13, you can also bind a StoreFront monitor to a service group. A monitor is bound to each member of the service group and probes are sent to the IP address and port of the bound member (service). Also, because each member of a service group is now monitored by using the member's IP address, you can now use the StoreFront monitor to monitor StoreFront cluster nodes that are added as members of the service group.

In earlier releases, the StoreFront monitor tried to authenticate anonymous stores. As a result, a service could be marked as DOWN and you could not launch XenApp or XenDesktop by using the URL of the load balancing virtual server.

From build 64.x, the probe order has changed. The monitor now determines the state of the StoreFront store by successively probing the account service, the discovery document, and then the authentication service, and skips authentication for anonymous stores.

The hostname parameter for StoreFront monitors is deprecated. The secure parameter is now used to determine whether to use HTTP (the default) or HTTPS to send monitor probes.

To use HTTPS, set the secure option to Yes.

To create a StoreFront monitor by using the command line interface

At the command prompt, type the following commands to configure a StoreFront monitor and verify the configuration:

```
add lb monitor <monitorName> STOREFRONT <string> -storeName <string> [-
    storefrontacctservice ( YES | NO )] -secure ( YES | NO )

show lb monitor <monitorName>
<!--NeedCopy-->
```

Example

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -
    storefrontacctservice YES -secure YES
<!--NeedCopy-->
```

To create a StoreFront monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type STOREFRONT.

> **Note**
>
> For more information about the StoreFront monitors, see Load balancing with NetScaler.

Custom monitors

September 21, 2020

In addition to built-in monitors, you can use custom monitors to check the state of your services. The NetScaler appliance provides several types of custom monitors based on scripts that are included with the NetScaler operating system that can be used to determine the state of services based on the load on the service or network traffic sent to the service. These are the inline monitors, user monitors, and load monitors.

With any of these types of monitors, you can use the supplied functionality, or you can create your own scripts and use those scripts to determine the state of the service to which the monitor is bound.

Configure HTTP-inline monitors

September 21, 2020

Inline monitors analyze and probe the responses from the services to which they are bound only when those services receive client requests. The inline monitor is of type HTTP-INLINE and can only be configured to work with HTTP and HTTPS services. An inline monitor determines that the service to which it is bound is UP by checking its responses to the requests that are sent to it. When no client requests are sent to the service, the inline monitor probes the service by using the configured URL.

Note: Inline monitors cannot be bound to HTTP or HTTPS Global Server Load Balancing (GSLB) remote or local services because these services represent virtual servers rather than actual load balanced Web servers.

Inline monitors have a time-out value and a retry count when probes fail. You can select any of the following action types for the NetScaler appliance to take when a failure occurs:

- **NONE.** No explicit action is taken. You can view the service and monitor, and the monitor indicates the number of current contiguous error responses and cumulative responses checked.
- **LOG.** Logs the event in ns/syslog and displays the counters.
- **DOWN.** Marks the service down and does not direct any traffic to the service. This setting breaks any persistent connections to the service. This action also logs the event and displays counters.

After the service is down, the service remains DOWN for the configured down time. After the DOWN time elapses, the inline monitor uses the configured URL to probe the service to see if it is available again. If the probe succeeds, the state of the service is changed to UP. Traffic is directed to the service, and monitoring resumes as before.

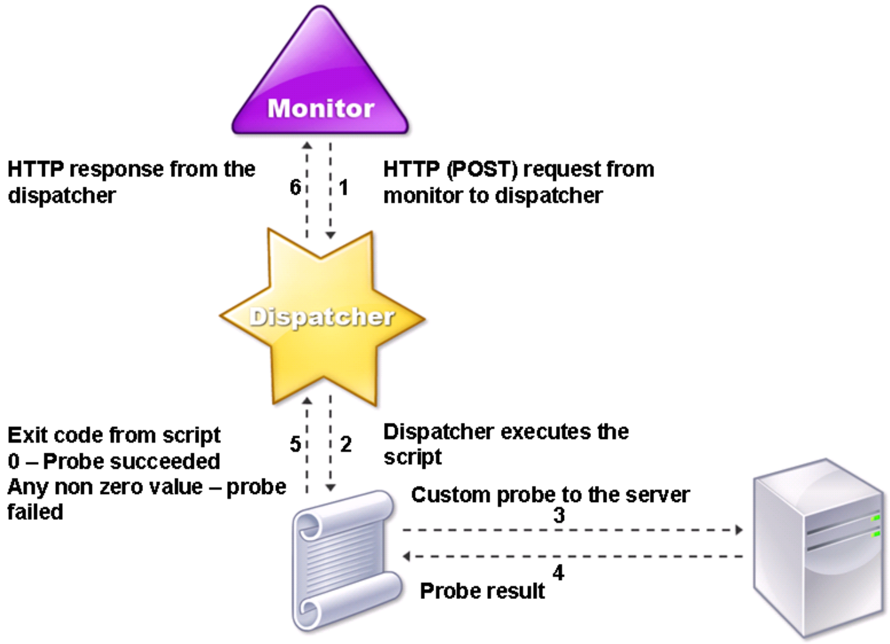
To configure inline monitors, see Configuring Monitors in a Load Balancing Setup.

Understand user monitors

September 21, 2020

User monitors extend the scope of custom monitors. You can create user monitors to track the health of customized applications and protocols that the NetScaler appliance does not support. The following diagram illustrates how a user monitor works.

Figure 1. User Monitors


Monitor
HTTP response from the dispatcher
6
1
HTTP (POST) request from monitor to dispatcher
Dispatcher
Exit code from script
0 – Probe succeeded
Any non zero value – probe failed
5
2
Dispatcher executes the script
Custom probe to the server
3
4
Probe result

A user monitor requires the following components.

- **Dispatcher.** A process, on the appliance, that listens to monitoring requests. A dispatcher can be on the loopback IP address (127.0.0.1) and port 3013. Dispatchers are also known as internal dispatchers. A dispatcher can also be a web server that supports Common Gateway Interface (CGI). Such dispatchers are also known as external dispatchers. They are used for custom scripts that do not run on the FreeBSD environment, such as .NET scripts.

  Note: You can configure the monitor and the dispatcher to use HTTPS instead of HTTP by enabling the "secure" option on the monitor and configure it as an external dispatcher. However, an internal dispatcher understands only HTTP, and cannot use HTTPS.

  In a HA setup, the dispatcher runs on both the primary and secondary NetScaler appliances. The dispatcher remains inactive on the secondary appliance.

Script. The script is a program that sends custom probes to the load balanced server and returns the response code to the dispatcher. The script can return any value to the dispatcher, but if a probe succeeds, the script must return a value of zero (0). The dispatcher considers any other value as probe failure.

The NetScaler appliance is bundled with sample scripts for commonly used protocols. The scripts exist in the /nsconfig/monitors directory. If you want to add a new script, add it there. If you want to customize an existing script, create a copy with a new name and modify it.

Important: Starting with release 10.1 build 122.17, the script files for user monitors are in a new location.

If you upgrade an MPX or VPX virtual appliance to release 10.1 build 122.17 or later, the changes are as follows:

- A new directory named conflicts is created in /nsconfig/monitors/ and all the built-in scripts of the previous builds are moved to this directory.
- All new built-in scripts are available in the /netscaler/monitors/ directory. All custom scripts are available in the /nsconfig/monitors/ directory.
- You must save a new custom script in the /nsconfig/monitors/ directory.
- After the upgrade is completed, if a custom script is created and saved in the /nsconfig/monitors/ directory, with the same name as that of a built-in script, the script in the /netscaler/monitors/ directory takes priority. That is, the custom script does not run.

If you provision a virtual appliance with release 10.1 build 122.17 or later, the changes are as follows:

- All built-in scripts are available in the /netscaler/monitors/ directory.
- The /nsconfig/monitors/ directory is empty.
- If you create a new custom script, you must save it in the /nsconfig/monitors/ directory.

For the scripts to function correctly:

- The maximum number of characters in the name of the script must not exceed 63.
- The maximum number of script arguments that can be provided to a script must not exceed 512.
- The maximum number of characters that can be provided in the parameter script arguments must not exceed 639.

To debug the script, you must run it by using the nsumon-debug.pl script from the CLI. You use the script name (with its arguments), IP address, and the port as the arguments of the nsumon-debug.pl script. Users must use the script name, IP address, port, time-out, and the script arguments for the nsumon-debug.pl script.

At the CLI, type:

```
nsumon-debug.pl <scriptname> <IP> <port> <timeout> <partitionID> [scriptarguments][is_secure]
<!--NeedCopy-->
```

Important: Starting with release 10.5 build 57.x, and 11.0 script files for user monitors support IPv6 addresses and include the following changes:

- For the following protocols, new pm files have been included for IPv6 support.
    - Radius

  - NNTP
  - POP3
  - SMTP

- The following sample scripts in /netscaler/monitors/ has been updated for IPv6 support:

  - nsbmradius.pl
  - nsldap.pl
  - nsnntp.pl
  - nspop3 nssf.pl
  - nssnmp.pl
  - nswi.pl
  - nstftp.pl
  - nssmtp.pl
  - nsrdp.pl
  - nsntlm-lwp.pl
  - nsftp.pl
  - nsappc.pl

  After upgrading to release 10.5 build 57.x, or 11.0, if you want to use your existing custom scripts with IPv6 services, make sure that you update the existing custom scripts with the changes provided in the updated sample scripts in
  /netscaler/monitors/.

  Note: The sample script nsmysql.pl does not support IPv6 address. If an IPv6 service is bound to a user monitor that uses nsmysql.pl, the probe will fail.

- The following LB monitor types have been updated to support IPv6 addresses:

  - USER
  - SMTP
  - NNTP
  - LDAP
  - SNMP
  - POP3
  - FTP_EXTENDED
  - STOREFRONT

  - APPC
  - CITRIX_WI_EXTENDED

  If you are creating a new custom script that uses one of these LB monitors types, make sure that you include IPv6 support in the custom script. Refer to the associated sample script in /netscaler/monitors/ for the changes that you have to make in the custom script for IPv6 support.

To track the status of the server, the monitor sends an HTTP POST request to the configured dispatcher. This POST request contains the IP address and port of the server, and the script that must be executed. The dispatcher executes the script as a child process, with user-defined parameters (if any). Then, the script sends a probe to the server. The script sends the status of the probe (response code) to the dispatcher. The dispatcher converts the response code to an HTTP response and sends it to the monitor. Based on the HTTP response, the monitor marks the service as up or down.

The appliance logs the error messages to the /var/nslog/nsumond.log file when user monitor probes fail. The following table lists the user monitors and the possible reasons for failure.

User monitor type	Probe failure reasons
SMTP	Monitor fails to establish a connection to the server.
NNTP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to find the NNTP group.
LDAP	Monitor fails to establish a connection to the server.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to bind to the LDAP server.
	Monitor fails to locate an entry for the target entity in the LDAP server.
FTP	The connection to the server times out.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.

User monitor type	Probe failure reasons
	Logon fails.
	Monitor fails to find the file on the server.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
POP3	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Preparation of SQL query fails.
	Execution of SQL query fails.
SNMP	Monitor fails to establish a connection to the database.
	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Logon fails.
	Monitor fails to create the SNMP session.
	Monitor fails to find the object identifier.
	The monitor threshold value setting is greater than or equal to the actual threshold of the monitor.
RDP (Windows Terminal Server)	Missing or invalid script arguments, which can include an invalid number of arguments or argument format.
	Monitor fails to create a socket.
	Mismatch in versions.

User monitor type	Probe failure reasons
	Monitor fails to confirm connection.

You can view the log file from the CLI by using the following commands, which open a BSD shell, display the log file on the screen, and then close the BSD shell and return you to the CLI:

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
<!--NeedCopy-->
```

User monitors also have a time-out value and a retry count for probe failures. You can use user monitors with non-user monitors. During high CPU utilization, a non-user monitor enables faster detection of a server failure.

If the user monitor probe times out during high CPU usage, the state of the service remains unchanged.

> **Note**
>
> For scriptable monitors, the response timeout must be configured to a value equal to expected timeout + 1 second.
>
> For example, if you expect the timeout to be 4 seconds, configure the response timeout as 5 seconds.
>
> **Example command:**

```
add lb monitor <name> USER - scriptname <script-name> -resptimeout 5
    seconds
<!--NeedCopy-->
```

How to use a user monitor to check web sites

September 21, 2020

You can configure a user monitor to check for specific Web site problems that are reported by HTTP servers using specific HTTP codes. The following table lists the HTTP response codes that this user monitor expects.

HTTP response code	Meaning
200 - success	Probe success.
503 - service unavailable	Probe failure.
404 - not found	Script not found or cannot execute.
500 - Internal server error	Internal error/resource constraints in dispatcher (out of memory, too many connections, unexpected system error, or too many processes). The service is not marked DOWN.
400 - bad request	Error parsing HTTP request.
502 - bad gateway	Error decoding script's response.

You configure the user monitor for HTTP by using the following parameters.

Parameter	Specifies
scriptName	The path and name of the script to execute.
scriptArgs	The strings that are added in the POST data. They are copied to the request verbatim.
dispatcherIP	The IP address of the dispatcher to which the probe is sent.
dispatcherPort	The port of the dispatcher to which the probe is sent.
localfileName	The name of a monitor script file on the local system.
destPath	A particular location on the NetScaler appliance where the uploaded local file is stored.

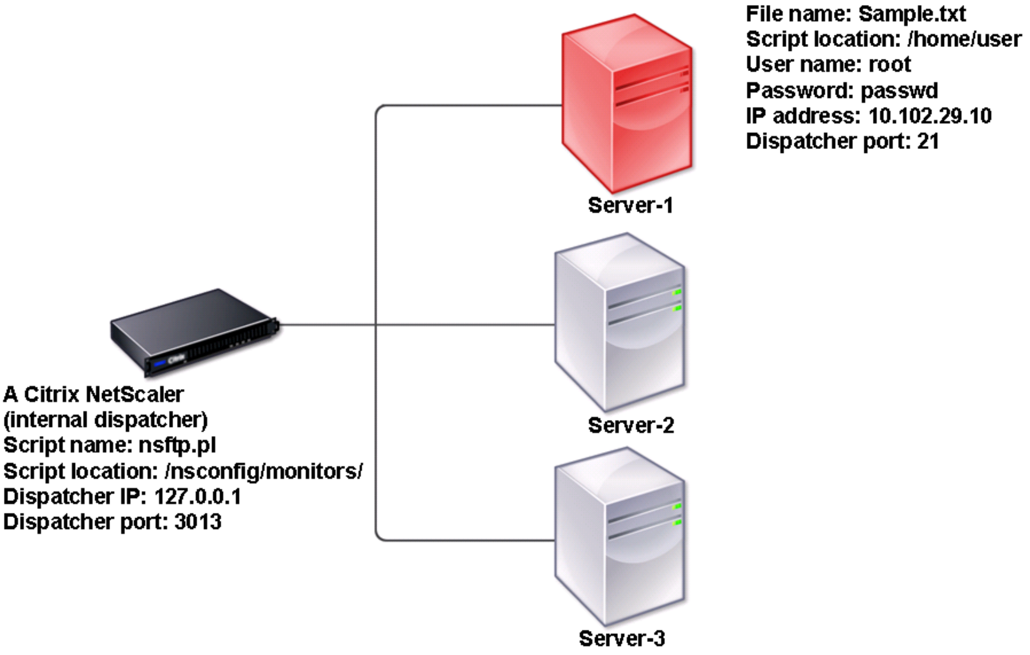
To create a user monitor to monitor HTTP, see Configuring Monitors in a Load Balancing Setup.

Understand the internal dispatcher

September 21, 2020

You can use a custom user monitor with the internal dispatcher. Consider a case where you need to track the health of a server based on the presence of a file on the server. The following diagram illustrates this scenario.

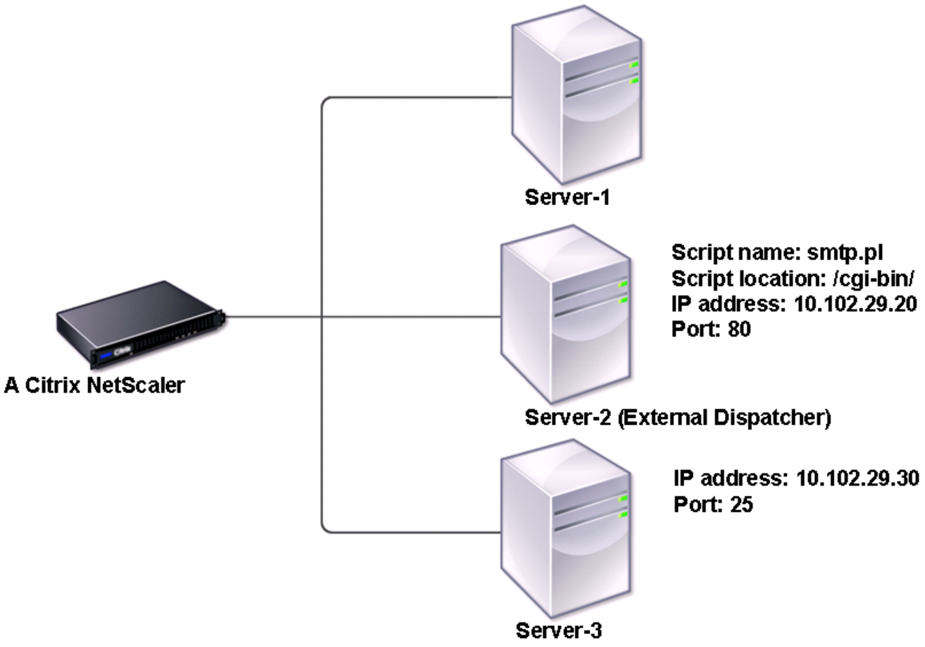
Figure 1. Using a user monitor with the internal dispatcher


File name: Sample.txt
Script location: /home/user
User name: root
Password: passwd
IP address: 10.102.29.10
Dispatcher port: 21
Server-1
Server-2
Server-3
A Citrix NetScaler
(internal dispatcher)
Script name: nsftp.pl
Script location: /nsconfig/monitors/
Dispatcher IP: 127.0.0.1
Dispatcher port: 3013

A possible solution is to use a Perl script that initiates an FTP session with the server and checks for the presence of the file. You can then create a user monitor that uses the Perl script. The NetScaler appliance includes such a Perl script (nsftp.pl), in the /nsconfig/monitors/ directory.

You can use a user monitor with an external dispatcher. Consider a case where you must track the health of a server based on the state of an SMTP service on another server. This scenario is illustrated in the following diagram.

Figure 2. Using a user monitor with an external dispatcher


Server-1
Script name: smtp.pl
Script location: /cgi-bin/
IP address: 10.102.29.20
Port: 80
A Citrix NetScaler
Server-2 (External Dispatcher)
IP address: 10.102.29.30
Port: 25
Server-3

A possible solution would be to create a Perl script that checks the state of the SMTP service on the server. You can then create a user monitor that uses the Perl script.

Configure a user monitor

September 21, 2020

To configure a user monitor, you must write a script that the monitor uses to check the services that are bound to it. Upload the script to the /nsconfig/monitors directory on the NetScaler appliance. Give executable permission to the script. If the monitor type is a protocol that the NetScaler appliance does not support, only then you must use the monitor of type **USER**. The user monitors support Perl and Bash scripts. Python scripts are not supported.

> **Note**
>
> Monitor probes originate from the NSIP address. The scriptargs configured for the monitor of type USER is displayed in the running configuration and ns.conf files.

To configure a user monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> USER -scriptname <NameOfScript> -
    scriptargs <Arguments>
<!--NeedCopy-->
```

Example:

```
add monitor Monitor-User-1 USER -scriptname nsftp.pl -scriptargs "file
    =/home/user/
sample.txt;user=root;password=passwd"
<!--NeedCopy-->
```

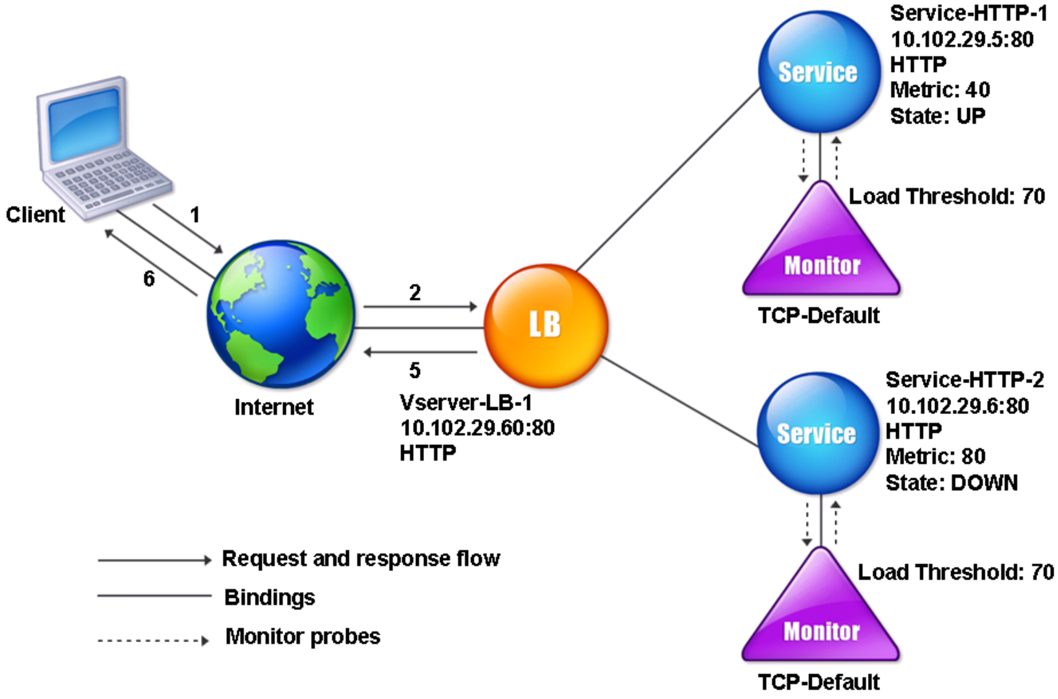
Understand load monitors

September 21, 2020

Load monitors use SNMP polled OIDs to calculate load. The load monitor uses the IP address of the service to which it is bound (the destination IP address) for polling. It sends an SNMP query to the service, specifying the OID for a metric. The metrics can be CPU, memory, or number of server connections. The server responds to the query with a metric value. The metric value in the response is compared with the threshold value. The NetScaler appliance considers the service for load balancing only if the metric is less than the threshold value. The service with the lowest load value is considered first.

The following diagram illustrates a load monitor configured for the services described in the basic load balancing setup discussed in Setting Up Basic Load Balancing.

Figure 1. Operation of Load Monitors


Client
1
6
Internet
2
5
LB
Vserver-LB-1
10.102.29.60:80
HTTP
Service
Service-HTTP-1
10.102.29.5:80
HTTP
Metric: 40
State: UP
Load Threshold: 70
Monitor
TCP-Default
Service
Service-HTTP-2
10.102.29.6:80
HTTP
Metric: 80
State: DOWN
Load Threshold: 70
Monitor
TCP-Default
Request and response flow
Bindings
Monitor probes

Note: The load monitor does not determine the state of the service. It only enables the appliance to consider the service for load balancing.

After you configure the load monitor, you must then configure the metrics that the monitor will use. For load assessment, the load monitor considers server parameters known as metrics, which are defined within the metric tables in the appliance configuration. Metric tables can be of two types:

- **Local.** By default, this table exists in the appliance. It consists of four metrics: connections, packets, response time, and bandwidth. The appliance specifies these metrics for a service, and SNMP queries are not originated for these services. These metrics cannot be changed.
- **Custom.** A user-defined table. Each metric is associated with an OID.

By default, the appliance generates the following tables:

- NetScaler
- RADWARE
- CISCO-CSS
- LOCAL
- FOUNDRY
- ALTEON

You can either add the appliance-generated metric tables, or you can add tables of your own choosing, as shown in the following table. The values in the metric table are provided only as examples. In an actual scenario, consider the real values for the metrics.

Metric name	OIDs	Weight	Threshold
CPU	1.2.3.4	2	70
Memory	4.5.6.7	3	80
Connections	5.6.7.8	4	90

To calculate the load for one or more metrics, you assign a weight to each metric. The default weight is 1. The weight represents the priority given to each metric. If the weight is high, the priority is high. The appliance chooses a service based on the SOURCEIPDESTIP hash algorithm.

You can also set the threshold value for each metric. The threshold value enables the appliance to select a service for load balancing if the metric value for the service is less than the threshold value. The threshold value also determines the load on each service.

Configure load monitors

September 21, 2020

To configure a load monitor, first create the load monitor. For instructions on creating a monitor, see Creating Monitors. Next, select or create the metric table to define a set of metrics that determine the state of the server, and (if you create a metric table) bind each metric to the metric table.

To create a metric table by using the command line interface

At the command prompt, type the following commands:

```
add lb metricTable <metricTableName>

bind lb metricTable <metricTableName> <metric> <SNMPOID>
<!--NeedCopy-->
```

Example:

```
add metricTable Table-Custom-1

bind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5 11
<!--NeedCopy-->
```

To create a metric table and bind metrics to it by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Metric Tables and create a metric table.
2. To bind metrics, click Bind and specify a metric and an SNMP OID.

Unbind metrics from a metrics table

September 21, 2020

You can unbind metrics from a metrics table if the metrics need to be changed, or if you want to remove the metrics table entirely.

To unbind metrics from a metric table by using the command line interface

At the command prompt, type:

```
unbind lb metricTable <metricTable> <metric>
<!--NeedCopy-->
```

Example:

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
<!--NeedCopy-->
```

To unbind metrics from a metric table by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Metric Tables**.
2. Open a metric table, select a metric, and click **Delete**.

You can view the detail of all configured metric tables, such as name and type, to determine whether the metric table is internal or created and configured.

Configure reverse monitoring for a service

September 21, 2020

A reverse monitor marks a service as DOWN if the probe criteria are satisfied and UP if they are not satisfied. For example, if you want a backup service to receive traffic only when the primary service is DOWN, you can bind a reverse monitor to the secondary service but configure it to probe the primary service.

The NetScaler appliance supports the following reverse monitors:

- HTTP
- ICMP
- TCP (from release 11.1 build 49.x)

Configuring HTTP Reverse Monitoring for a Service

The following table describes the conditions of HTTP direct and reverse monitoring for a service:

Condition	Direct	Reverse
Connection not established.	Fail	Fail
HTTP response code matches the probe’s specifications.	Success	Fail
HTTP response code does not match the probe’s specifications.	Fail	Success
Probe timed out.	Fail	Fail

To configure HTTP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> HTTP -respCode 200 -httpRequest "HEAD /" -destIP <Primary_Service_IP_Address> -destPort 80 -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

Configuring ICMP Reverse Monitoring for a Service

The following table describes the conditions of ICMP direct and reverse monitoring for a service:

Condition	Direct	Reverse
ICMP echo reply is received.	Success	Fail

Probe timed out.	Fail	Success

To configure ICMP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> PING -destIP <Primary_Service_IP_Address> -reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

Configuring TCP Reverse Monitoring for a Service

If a direct TCP monitor receives a RESET in response to a monitor probe, the service is marked DOWN. However, if a reverse TCP monitor receives a RESET response, the probe is considered successful, and the service is marked UP.

The following table describes the conditions of TCP reverse monitoring for a service:

Condition	**Direct**	**Reverse**
TCP connection is established.	Success	Fail
Probe timed out.	Fail	Fail
Response to probe is RESET.	Fail	Success

To configure TCP reverse monitoring for a service by using the CLI

At the command prompt, type:

```
add lb monitor <Monitor_Name> TCP - destip <Primary_Service_IP_Address> -destport <primary_service_port> - reverse YES

bind service <Secondary_Service_Name> -monitorname <Monitor_Name>
<!--NeedCopy-->
```

To configure reverse monitoring by using the GUI

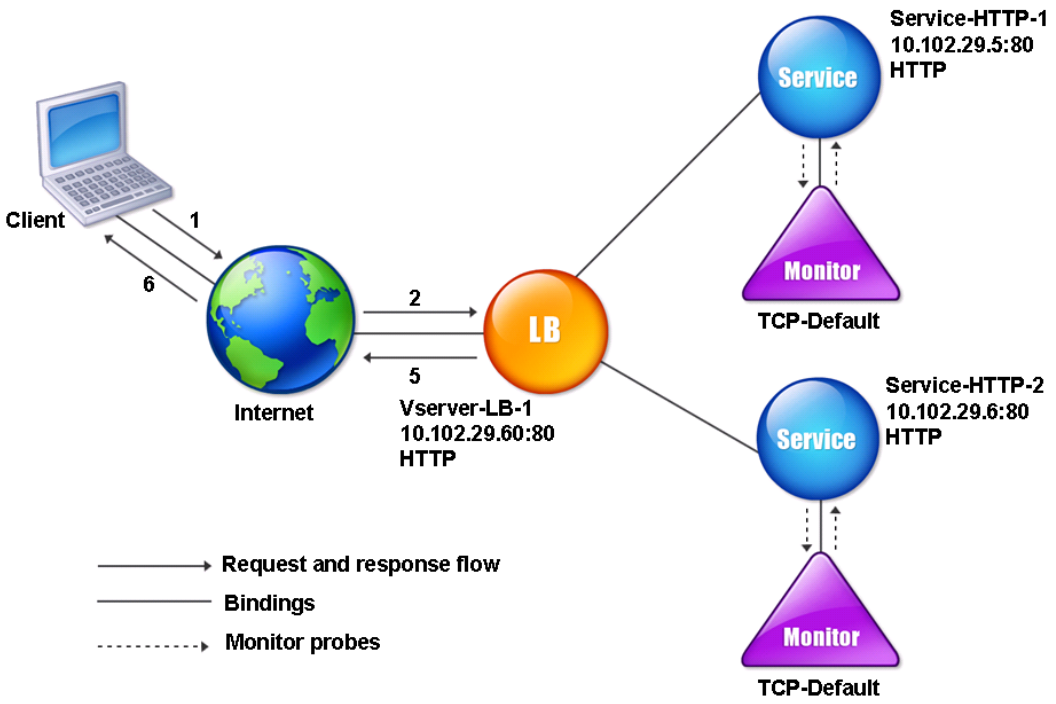
1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Create an HTTP, ICMP, or TCP monitor and select **Reverse**.

Configure monitors in a load balancing setup

September 21, 2020

To configure monitors on a website, you first decide whether to use a built-in monitor or create your own monitor. If you create a monitor, you can choose between creating a monitor based on a built-in monitor, or creating a custom monitor that uses a script that you write to monitor the service. For more information about creating custom monitors, see Custom Monitors. Once you have chosen or created a monitor, you then bind it to the appropriate service. The monitor names can be up to 63 characters in length. The following conceptual diagram illustrates a basic load balancing setup with monitors.

Figure 1. How monitors operate


Service-HTTP-1
10.102.29.5:80
HTTP
Service
Client
1
6
Monitor
TCP-Default
2
LB
5
Internet
Vserver-LB-1
10.102.29.60:80
HTTP
Service-HTTP-2
10.102.29.6:80
HTTP
Service
Request and response flow
Bindings
Monitor probes
Monitor
TCP-Default

As shown in the diagram, each service has a monitor bound to it. The monitor probes the load balanced server via its service. As long as the load balanced server responds to the probes, the monitor marks it UP. If the load balanced server fails to respond to the designated number of probes within the designated time period, the monitor marks it DOWN.

Create monitors

September 21, 2020

The NetScaler appliance provides a set of built-in monitors. It also allows you to create custom monitors, either based on the built-in monitors or from scratch.

To create a monitor by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> [<interval>]
<!--NeedCopy-->
```

Example:

```
add lb mon monitor-HTTP-1 HTTP

add lb mon monitor-HTTP-2 TCP 2
<!--NeedCopy-->
```

To create a monitor by using the GUI

1. Navigate to **Traffic Management**> **Load Balancing**> **Monitors**.
2. Click **Add** and create a monitor type that meets your requirement.

Bind monitors to services

September 21, 2020

After creating a monitor, you bind it to a service. You can bind one or multiple monitors to a service. If you bind one monitor to a service, that monitor determines whether the service is marked UP or DOWN.

If you bind multiple monitors to a service, the NetScaler appliance checks the state of all the monitors and then decides the state of the service. You can configure different weights to a monitor. The weight of a monitor specifies how much that monitor contributes to designating the service as UP or DOWN. A monitor with a greater weight has a higher preference in marking the service UP or DOWN. Default weight is 1. Therefore, even if one of the monitors fails, the service is marked as DOWN. For more information, see Set a threshold value for the monitors bound to a service.

Note: The destination IP address of a monitor probe can be different than the server IP address and port.

To bind a monitor to a service by using the command line interface

At the command prompt, type:

```
bind service <name> (-monitorName <string>)
<!--NeedCopy-->
```

Example:

```
bind service s1 -monitorName tcp
<!--NeedCopy-->
```

To bind a monitor to a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and add a monitor.

Modify monitors

September 21, 2020

You can modify the settings for any monitor that you created.

Note: Two sets of parameters apply to monitors: those that apply to all monitors, regardless of type, and those that are specific to a monitor type. For information on parameters for a specific monitor type, see the description for that type of monitor.

To modify an existing monitor by using the command line interface

At the command prompt, type:

```
set lb monitor <monitorName> <type> -interval <interval> -resptimeout <resptimeout>
<!--NeedCopy-->
```

Example:

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
<!--NeedCopy-->
```

To modify an existing monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and open a monitor to modify.

Enable and disable monitors

September 21, 2020

By default, monitors bound to services and service groups are enabled. When you enable a monitor, the monitor begins probing the services to which it is bound. If you disable a monitor bound to a service, the state the service is determined using the other monitors bound to the service. If the service is bound to only one monitor, and if you disable the monitor, the state of the service is determined using the default monitor.

To enable a monitor by using the command line interface

At the command prompt, type:

```
enable lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
enable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

To enable a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and from the Action list, select Enable or Disable.

To disable a monitor by using the command line interface

At the command prompt, type:

```
disable lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
disable lb mon monitor-HTTP-1
<!--NeedCopy-->
```

Unbind monitors

September 21, 2020

You can unbind monitors from a service and service group. When you unbind a monitor from the service group, the monitors are unbound from the individual services that constitute the service group. When you unbind a monitor from a service or a service group, the monitor does not probe the service or the service group.

Note: When you unbind all user-configured monitors from a service or a service group, the default monitor is bound to the service and the service group. The default monitors then probes the service or the service groups.

To unbind a monitor from a service by using the command line interface

At the command prompt, type:

```
unbind lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
unbind mon monitor-HTTP-1 Service-HTTP-1
<!--NeedCopy-->
```

To unbind a monitor from a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service to modify.
2. Click in the Monitors section, select a monitor, and click Unbind.

Remove monitors

September 21, 2020

After you unbind a monitor that you created from its service, you can remove that monitor from the NetScaler configuration. (If a monitor is bound to a service, it cannot be removed.)

Note: When you remove monitors bound to a service, the default monitor is bound to the service. You cannot remove default monitors.

To remove a monitor by using the command line interface

At the command prompt, type:

```
rm lb monitor <monitorName> <type>
<!--NeedCopy-->
```

Example:

```
rm lb monitor monitor-HTTP-1 HTTP
<!--NeedCopy-->
```

To remove a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and click Delete.

View monitors

September 21, 2020

You can view the services and service groups that are bound to a monitor. You can verify the settings of a monitor to troubleshoot your NetScaler configuration. The following procedure describes the steps to view the bindings of a monitor to the services and service groups.

To view monitor bindings by using the command line interface

At the command prompt, type:

```
show lb monbindings <MonitorName>
<!--NeedCopy-->
```

Example:

```
show lb monbindings monitor-HTTP-1
<!--NeedCopy-->
```

To view monitor bindings by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and in the Action list, click Show Bindings.

To view monitors by using the command line interface

At the command prompt, type:

```
show lb monitor <monitorName>
<!--NeedCopy-->
```

Example:

```
show lb mon monitor-HTTP-1
<!--NeedCopy-->
```

To view monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors. The details of the available monitors appear in the Monitors pane.

Close monitor connections

September 21, 2020

The NetScaler appliance sends probes to the services through the monitors bound to the services. By default, the monitor on the appliance and the physical server follow the complete handshake procedure even for monitor probes. However, this procedure adds overhead to the monitoring process and may not be always necessary.

For the TCP type monitor, you can configure the appliance to close a monitor-probe connection after receiving SYN-ACK from the service. To do so, set the value of the monitorConnectionClose parameter to RESET. If you want the monitor-probe connection to go through the complete procedure, set the value to FIN.

Note: The monitorConnectionClose setting is applicable only for type TCP and TCP-Default monitors.

To configure monitor-connection closure by using the command line interface:

At the command prompt, type:

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
<!--NeedCopy-->
```

Example

```
set lb parameter -monitorConnectionClose RESET
<!--NeedCopy-->
```

To configure monitor-connection closure by using the configuration utility:

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters.
2. Select FIN or Reset.

Closing Monitor Connections at the Service or Service Group Level

You can also configure the appliance to close a monitor-probe connection at the service and service group level by setting the monConnectionClose parameter. If this parameter is not set, the monitor connection is closed by using the value set in the global load balancing parameters. If this parameter is set at the service or service group level, the monitor connection is closed by sending a connection termination message, with the FIN or RESET bit set, to the service or service group.

To configure monitor-connection closure at the service level by using the CLI

At the command prompt, type:

```
set service <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

To configure monitor-connection closure at the service group level by using the CLI

At the command prompt, type:

```
 set serviceGroup <service_name> -monConnectionClose ( RESET | FIN )
<!--NeedCopy-->
```

To configure monitor-connection closure at the service level by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Add or edit a service, and in **Basic Settings**, set the **Monitoring Connection Close Bit**.

To configure monitor-connection closure at the service group level by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Add or edit a service group, and in **Basic Settings**, set the **Monitoring Connection Close Bit**.

Ignore the upper limit on client connections for monitor probes

September 21, 2020

Depending on considerations such as the capacity of a physical server, you can specify a limit on the maximum number of client connections made to any service. If you have set such a limit on a service, the NetScaler appliance stops sending requests to the service when the threshold is reached and resumes sending connections to the service after the number of existing connections falls to within the limits. You can configure the appliance to skip this check when it sends monitor-probe connections to a service.

Note: You cannot skip the maximum-client-connections check for an individual service. If you specify this option, it applies to all the monitors bound to all the services configured on the NetScaler appliance.

To set the Skip MaxClients for Monitor Connections option by using the command line interface

At the command prompt, type:

```
set lb parameter -monitorSkipMaxClient (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
set lb parameter -monitorSkipMaxClient enabled
<!--NeedCopy-->
```

To set the Skip MaxClients for Monitor Connections option by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters.
2. Select Skip MaxClients for Monitoring Connections.

Manage a large scale deployment

September 21, 2020

The NetScaler appliance contains several features that are helpful when you are configuring a large load balancing deployment. Instead of configuring virtual servers and services individually, you can create groups of virtual servers and services. You can also create a range of virtual servers and services, and you can translate or mask virtual server and service IP addresses.

You can set persistence for a group of virtual servers. You can bind monitors to a group of services. Creating a range of virtual servers and services of identical type allows you to set up and configure those servers in a single procedure, which significantly shortens the time required to configure those virtual servers and services.

By translating or masking IP addresses, you can take down virtual servers and services, and make changes to your infrastructure, without extensive reconfiguration of your service and virtual server definitions.

Ranges of virtual servers and services

September 21, 2020

When you configure load balancing, you can create ranges of virtual servers and services, eliminating the need to configure virtual servers and services individually. For example, you can use a single procedure to create three virtual servers with three corresponding IP addresses. When more than one argument uses a range, all of the ranges must be of the same size.

The following are the types of ranges you can specify when adding services and virtual servers to your configuration:

- **Numeric ranges.** Instead of typing a single number, you can specify a range of consecutive numbers.

  For example, you can create a range of virtual servers by specifying a starting IP address, such as 10.102.29.30, and then typing a value for the last byte that indicates the range, such as 34. In this example, five virtual servers will be created with IP addresses that range between 10.102.29.30 and 10.102.29.34.

Note: The IP addresses of the virtual servers and services must be consecutive.

- **Alphabetic ranges.** Instead of typing a literal letter, you can substitute a range for any single letter, for example, [C-G]. This results in all letters in the range being included, in this case C, D, E, F, and G.

  For example, if you have three virtual servers named Vserver-x, Vserver-y, and Vserver-z, instead of configuring them separately, you can type vserver [x-z] to configure them all.

Creating a range of virtual servers

You create a range of virtual servers as described below.

To create range of virtual servers by using the CLI

At the command prompt, type one of the following commands:

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<
    port>]

add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue
    >]> [<port>]
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

OR

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80

vserver "vserverP" added

vserver "vserverQ" added

vserver "vserverR" added

Done
<!--NeedCopy-->
```

To create range of virtual servers by using the command line interface

At the command prompt, type one of the following commands:

```
add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<port>]

add lb vserver <name>@**[**<rangeValue>**]**> <protocol> <IPAddress[<rangeValue>]> [<port>]
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
<!--NeedCopy-->
```

OR

```
add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
Done
<!--NeedCopy-->
```

To create range of virtual servers by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Add a virtual server, and specify a range.

Creating a range of services

You create a range of services as described below. If you specify a range for the service name, specify a range for the IP address too.

To create range of services by using the command line interface

At the command prompt, type the command:

```
add service <name>@ <IP>@ <protocol> <port>
<!--NeedCopy-->
```

Example:

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
 Done
<!--NeedCopy-->
```

Configure service groups

September 21, 2020

Configuring a service group enables you to manage a group of services as easily as a single service. For example, if you enable or disable any option, such as compression, health monitoring or graceful shutdown, for a service group, the option gets enabled for all the members of the service group.

After creating a service group, you can bind it to a virtual server, and you can add services to the group. You can also bind monitors to service groups.

The members of a service group can be identified by IP address or server name.

Using domain-name based service (DBS) group members is advantageous because you need not reconfigure the member on the NetScaler appliance if the IP address of the member changes. The appliance automatically senses such changes through the configured name server. This feature is particularly useful in cloud scenarios, where the service provider can change a physical server or change the IP address for a service. If you specify a DBS group member, the appliance learns the IP address dynamically.

You can bind both IP-based and DBS members to the same service group.

Note: If you use DBS service group members, make sure that either a name server is specified or a DNS server is configured on the NetScaler appliance. A domain name will be resolved into an IP address only if the corresponding address record is present on the appliance or the name server.

Create service groups

You can configure up to 8192 service groups on the NetScaler appliance.

To create a service group by using the command line

At the command prompt, type:

```
add servicegroup <ServiceGroupName> <Protocol>
<!--NeedCopy-->
```

Example:

```
add servicegroup Service-Group-1 HTTP
<!--NeedCopy-->
```

To create a service group by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**, and add a service group.

Bind a service group to a virtual server

When you bind a service group to a virtual server, the member services are bound to the virtual server.

To bind a service group to a virtual server by using the command line interface

At the command prompt, type:

```
bind lb vserver <name>@ <serviceGroupName>
<!--NeedCopy-->
```

Example:

bind lb vserver Vserver-LB-1 Service-Group-1

To bind a service group to a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In Advanced Settings, select **Service Groups**.

Bind a member to a service group

Adding services to a service group enables the service group to manage the servers. You can add the servers to a service group by specifying the IP addresses or the names of the servers.

In the configuration utility, if you want to add a domain-name based service group member, select Server Based.

With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.

To add members to a service group by using the command line interface

To configure a service group, at the command prompt, type:

```
bind servicegroup <serviceGroupName> (<IP>@ | <serverName>) <port>
<!--NeedCopy-->
```

Examples:

```
bind servicegroup Service-Group-1 10.102.29.30 80

bind servicegroup Service-Group-2 1000:0000:0000:0000:0005:0600:700a
    :888b 80

bind servicegroup CitrixEdu s1.citrite.net
<!--NeedCopy-->
```

To add members to a service group by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groupsand open a service group.
2. Click in the Service Group section, and do one of the following:
    - To add a new IP based service group member, select IP Based.
    - To add a server-name based service group member, select Server Based.

    If you want to add a domain-name based service group member, select **Server Based**. With this option, you can add any server that has been assigned a name, regardless of whether the name is an IP address or a user-assigned name.
3. If adding a new IP based member, in the IP Address text box, type the IP address. If the IP address uses IPv6 format, select the IPv6 check box and then enter the address in the IP Address text box

    Note: You can add a range of IP addresses. The IP addresses in the range must be consecutive. Specify the range by entering the starting IP address in the IP Address text box (for example, 10.102.29.30). Specify the end byte of the IP address range in the text box under Range (for example, 35). In the Port text box type the port (for example, 80), and then click Add.
4. Click Create.

Bind a monitor to a service group

When you create a service group, the default monitor of the type appropriate for the group is automatically bound to it. Monitors periodically probe the servers in the service group to which they are bound and update the state of the service groups.

You can bind a different monitor of your own choice to the service group.

To bind a monitor to a service group by using the command line interface

At the command prompt, type:

```
bind serviceGroup <serviceGroupName> -monitorName <string> -monState (ENABLED | DISABLED)
<!--NeedCopy-->
```

Example:

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

To a bind monitor to a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group and, in Advanced Settings, click **Monitors**.

Retain the original state of a service group member after disabling and enabling a virtual server

From build 64.x, a new global option, –retainDisableServer, enables you to retain a service-group member's state when a server is disabled and reenabled.

Previously, a member's state would change from DISABLED to ENABLED under the following set of conditions:

- Two applications are deployed on the same port on a virtual server.
- Two service groups with a common member are bound to this virtual server, and the common member is enabled in one group and disabled in the other.
- The server is disabled and then reenabled.

Under these conditions, disabling the server disables all the service group members, and reenabling the server enables all the members, by default, regardless of their earlier states. To bring the members back to the original states, you must manually disable those member(s) in the service group. This is a cumbersome task and prone to errors.

Manage service groups

September 21, 2020

You can change the settings of the services in a service group, and you can perform tasks such as enabling, disabling, and removing service groups. You can also unbind members from a service group. For more information about service groups, see Configure service groups.

Modify a service group

You can modify attributes of service group members. You can set several attributes of the service group, such as maximum client, SureConnect, and compression. The attributes are set on the individual servers in the service group. You cannot set parameters on the service group such as transport information (IP address and port), weight, and server ID.

Note: A parameter you set for a service group is applied to the member servers in the group, not to individual services.

To modify a service group by using the command line interface

At the command prompt, type the following command with one or more of the optional parameters:

```
set servicegroup <serviceGroupName> [-type <type>] [-maxClient <maxClient>] [-maxReq <maxReq>] [-cacheable (YES|NO)] [-cip (ENABLED|DISABLED)] [-cipHeader <cipHeader>] [-usip (YES|NO)] [-sc (ON|OFF)] [-sp (ON|OFF)] [-cltTimeout <cltTimeout>] [-svrTimeout <svrTimeout>] [-cka (YES|NO)] [-TCPB (YES|NO)] [-CMP (**YES**|**NO**)] [-maxBandwidth <maxBandwidth>] [-maxThreshold <maxThreshold>] [-state (ENABLED|DISABLED)] [-downStateFlush (ENABLED|DISABLED)
<!--NeedCopy-->
```

Example:

```
set servicegroup Service-Group-1 -type TRANSPARENT

set servicegroup Service-Group-1 -maxClient 4096

set servicegroup Service-Group-1 -maxReq 16384

set servicegroup Service-Group-1 -cacheable YES
<!--NeedCopy-->
```

To modify a service group by using the configuration utility

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**, and open the service group to modify.

Remove a service group

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the NetScaler appliance.

To remove a service group by using the command line interface

At the command prompt, type:

```
rm servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
rm servicegroup Service-Group-1
<!--NeedCopy-->
```

To remove a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and click **Delete**.

Unbind a member from a service group

When you unbind a member from the service group, the attributes set on the service group no longer apply to the member that you unbound. The member services retain its individual settings, however, and continue to exist on the NetScaler appliance.

To unbind members from a service group by using the command line interface

At the command prompt, type:

```
unbind servicegroup <serviceGroupName> <IP>@ [<port>]
<!--NeedCopy-->
```

Example:

```
unbind servicegroup Service-Group-1 10.102.29.30 80
<!--NeedCopy-->
```

To unbind members from a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group, and click in the Service Group Members section.
3. Select a service group member, and click **Unbind**.

Unbind a service group from a virtual server

When you unbind a service group from a virtual server, the member services are unbound from the virtual server and continue to exist on the NetScaler appliance.

To unbind a service group from a virtual server by using the command line interface

At the command prompt, type:

```
unbind lb vserver <name>@ <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
unbind lb vserver Vserver-LB-1 Service-Group-1
<!--NeedCopy-->
```

To unbind a service group from a virtual server by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open the virtual server, and click in the Service Group section.
3. Select the service group, and click **Unbind**.

Unbind monitors from service groups

When you unbind a monitor from a service group, the monitor that you unbound no longer monitors the individual services that constitute the group.

To unbind a monitor from a service group using the command line interface

At the command prompt, type:

```
unbind serviceGroup <serviceGroupName> -monitorName <string>
<!--NeedCopy-->
```

Example:

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
<!--NeedCopy-->
```

To unbind a monitor from a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Open a service group, and click in the Monitors section.
3. Select a monitor, and click **Unbind**.

Enable or Disable a service group

When you enable a service group and the servers, the services belonging to the service group are enabled. Similarly, when a service belonging to a service group is enabled, the service group and the service are enabled. By default, service groups are enabled.

After disabling an enabled service, you can view the service using the configuration utility or the command line to see the amount of time that remains before the service goes DOWN.

To disable a service group by using the command line interface

At the command prompt, type:

```
disable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
disable servicegroup Service-Group-1
<!--NeedCopy-->
```

To disable a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Disable**.

To enable a service group by using the command line interface

At the command prompt, type:

```
enable servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
enable servicegroup Service-Group-1
<!--NeedCopy-->
```

To enable a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Enable**.

View the status of service groups members

Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.

Viewing the properties of a service group

You can view the following settings of the configured service groups:

- Name
- IP address
- State
- Protocol
- Maximum client connections
- Maximum requests per connection
- Maximum bandwidth
- Monitor threshold

Viewing the details of the configuration can be helpful for troubleshooting your configuration.

To view the properties of a service group by using the command line interface

At the command prompt, type one of the following commands to display the group properties or the properties and the group members:

```
show servicegroup <ServiceGroupName>

show servicegroup <ServiceGroupName> -includemembers
<!--NeedCopy-->
```

Example:

```
show servicegroup Service-Group-1
<!--NeedCopy-->
```

To view the properties of a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Click the arrow next to the service group.

Viewing service group statistics

You can view service-group statistical data, such as rate of requests, responses, request bytes, and response bytes. The NetScaler appliance uses the statistics of a service group to balance the load on the services.

To view the statistics of a service group by using the command line interface

At the command prompt, type:

```
stat servicegroup <ServiceGroupName>
<!--NeedCopy-->
```

Example:

```
stat servicegroup Service-Group-1
<!--NeedCopy-->
```

To view the statistics of a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and click **Statistics**.

Load balancing virtual servers bound to a service group

In large-scale deployments, the same service group can be bound to multiple load balancing virtual servers. In such a case, instead of viewing each virtual server to see the service group it is bound to, you can view a list of all the load balancing virtual servers bound to a service group. You can view the following details of each virtual server:

- Name
- State
- IP address
- Port

To display the virtual servers bound to a service group by using the command line interface

At the command prompt, type the following command to display the virtual servers bound to a service group:

```
show servicegroupbindings <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
> show servicegroupbindings SVCGRPDTLS
SVCGRPDTLS - State :ENABLED
1) Test-pers (10.10.10.3:80) - State : DOWN
2) BRVSERV (10.10.1.1:80) - State : DOWN
3) OneMore (10.102.29.136:80) - State : DOWN
4) LBVIP1 (10.102.29.66:80) - State : UP
Done
>
<!--NeedCopy-->
```

To display the virtual servers bound to a service group by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Select a service group, and in the Action list, click **Show Bindings**.

Configure automatic domain based service group scaling

September 21, 2020

A domain based service group consists of members whose IP addresses are obtained by resolving the domain names of servers that are bound to the service group. The domain names are resolved by a name server whose details you configure on the appliance. A domain based service group can also include IP-address based members.

The process of name resolution for a domain based server might return more than one IP address. The number of IP addresses in the DNS response is determined by the number of address (A) records configured for the domain name, on the name server. Even if the name resolution process returns multiple IP addresses, only one IP address is bound to the service group. To scale up or scale down a service group, you need to manually bind and unbind additional domain based servers to and from the service group, respectively.

However, you can configure a domain based service group to scale automatically on the basis of the complete set of IP addresses returned by a DNS name server for a domain based server. To configure automatic scaling, when binding a domain based server to a service group, enable the automatic scaling option. Following are the steps for configuring a domain based service group that scales automatically:

- Add a name server for resolving domain names. For more information about configuring a name server on the appliance, see Adding a Name Server.
- Add a domain based server. For information about adding a domain based server, see Adding a Server.
- Add a service group and associate the domain based server to the service group, with the autoscale option set to DNS. For information about adding a service group, see Configuring Service Groups.

When a domain based server is bound to a service group and the automatic scaling option is set on the binding, a UDP monitor and a TCP monitor are automatically created and bound to the domain based server. The two monitors function as resolvers. The TCP monitor is disabled by default, and the appliance uses the UDP monitor to send DNS queries to the name server to resolve the domain name. If the DNS response is truncated (has the TC flag set to 1), the appliance falls back to TCP and uses the TCP monitor to send the DNS queries over TCP. Thereafter, the appliance continues to use only the TCP monitor.

The DNS response from the name server might contain multiple IP addresses for the domain name. With the automatic scaling option set, the appliance polls each of the IP addresses by using the default monitor, and then includes in the service group only those IP addresses that are up and available. After the IP address records expire, as defined by their time-to-live (TTL) values, the UDP monitor (or the TCP monitor, if the appliance has fallen back to using the TCP monitor) queries the name server for domain resolution and includes any new IP addresses in the service group. If an IP address that is part of the service group is not present in the DNS response, the appliance removes that address from the service group after gracefully closing existing connections to the group member, a process during which it does not allow any new connections to be established with the member. If a domain name that resolved successfully in the past results in an NXDOMAIN response, all the service group members associated with that domain are removed.

Static (IP-address based) members and dynamically scaling domain based members can coexist in a service group. You can also bind members with different domain names to a service group with the automatic scaling option set. However, each domain name associated with a service group must be unique within the service group. You must enable the automatic scaling option for each domain based server that you want to use for automatic service group scaling. If an IP address is common to one or more domains, the IP address is added to the service group only once.

To configure a service group to scale automatically by using the command line interface

At the command prompt, type the following commands to configure the service group and verify the configuration:

```
add serviceGroup <serviceGroupName> -autoScale (YES | NO)

show serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example

In the following example, server1 is a domain based server. The DNS response contains multiple IP addresses. Five addresses are available and are added to the service group.

```
> add serviceGroup servGroup server1 80 -autoScale YES
 Done
> sh servicegroup servGroup
       servGroup - HTTP
        State: ENABLED  Monitor Threshold : 0
              . . .
              . . .
        1)   192.0.2.31:80   State: UP       Server Name: server1 (Auto scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        2)   192.0.2.32:80   State: UP       Server Name: server1 (Auto scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        3)   192.0.2.36:80   State: UP       Server Name: server1 (Auto scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        4)   192.0.2.55:80   State: UP       Server Name: server1 (Auto scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.

        5)   192.0.2.80:80   State: UP       Server Name: server1 (Auto scale)    Server ID: None Weight: 1

                Monitor Name: tcp-default       State: UP
                Probes: 2       Failed [Total: 0 Current: 0]
                Last response: Success - TCP syn+ack received.
 Done
<!--NeedCopy-->
```

To configure a service group to scale automatically by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**.
2. Create a service group, and set the autoscale mode to DNS.

Translate the IP address of a domain-based server

September 21, 2020

To simplify maintenance on the NetScaler appliance and on the domain-based servers that are connected to it, you can configure IP address masks and translation IP addresses. These functions work together to parse incoming DNS packets and substitute a new IP address for a DNS-resolved IP address.

When configured for a domain-based server, IP address translation enables the appliance to locate an alternate server IP address in the event that you take the server down for maintenance or if you make any other infrastructure changes that affect the server.

When configuring the mask, you must use standard IP mask values (a power of two, minus one) and zeros, for example, 255.255.0.0. Non-zero values are only permitted in the starting octets.

When you configure a translation IP for a server, you create a 1:1 correspondence between a server IP address and an alternate server that shares leading or trailing octets in its IP address. The mask blocks particular octets in the original server's IP address. The DNS-resolved IP address is transformed to a new IP address by applying the translation IP address and the translation mask.

For example, you can configure a translation IP address of 10.20.0.0 and a translation mask of 255.255.0.0. If a DNS-resolved IP address for a server is 40.50.27.3, this address is transformed to 10.20.27.3. In this case, the translation IP address supplies the first two octets of the new address, and the mask passes through the last two octets from the original IP address. The reference to the original IP address, as resolved by DNS, is lost. Monitors for all services to which the server is bound also report on the transformed IP address.

When configuring a translation IP address for a domain-based server, you specify a mask and an IP address to which the DNS-resolved IP address is to be translated.

Note: Translation of the IP address is only possible for domain-based servers. You cannot use this feature for IP-based servers. The address pattern can be based on IPv4 addresses only.

To configure a translation IP address for a server by using the command line interface

At the command prompt, type:

```
add server <name>@ <serverDomainName> -translationIp <translationIPAddress> -translationMask <netMask> -state <ENABLED|DISABLED>
<!--NeedCopy-->
```

Example:

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -translationMask
255.255.0.0 -state ENABLED
<!--NeedCopy-->
```

To configure a translation IP address for a server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Servers, create a domain-based server, and specify a translation IP address.

Mask a virtual server IP address

September 21, 2020

You can configure a mask and a pattern instead of a fixed IP address for a virtual server. This enables traffic that is directed to any of the IP addresses that match the mask and pattern to be rerouted to a particular virtual server. For example, you can configure a mask that allows the first three octets of an

IP address to be variable, so that traffic to 111.11.11.198, 22.22.22.198, and 33.33.33.198 is all sent to the same virtual server.

By configuring a mask for a virtual server IP address, you can avoid reconfiguration of your virtual servers due to a change in routing or another infrastructure change. The mask allows the traffic to continue to flow without extensive reconfiguration of your virtual servers.

The mask for a virtual server IP address works somewhat differently from the IP pattern definition for a server described in Translating the IP Address of a Domain-Based Server. For a virtual server IP address mask, a non-zero mask is interpreted as an octet that is considered. For a service, the non-zero value is blocked.

Additionally, for a virtual server IP address mask, either leading or trailing values can be considered. If the virtual server IP address mask considers values from the left of the IP address, this is known as a forward mask. If the mask considers the values to the right side of the address, this is known as a reverse mask.

Note: The NetScaler appliance evaluates all forward mask virtual servers before evaluating reverse mask virtual servers.

When masking a virtual server IP address, you also need to create an IP address pattern for matching incoming traffic with the correct virtual server. When the appliance receives an incoming IP packet, it matches the destination IP address in the packet with the bits that are considered in the IP address pattern, and after it finds a match, it applies the IP address mask to construct the final destination IP address.

Consider the following example:

- Destination IP address in the incoming packet: 10.102.27.189
- IP address pattern: 10.102.0.0
- IP mask: 255.255.0.0
- Constructed (final) destination IP address: 10.102.27.189.

In this case, the first 16 bits in the original destination IP address match the IP address pattern for this virtual server, so this incoming packet is routed to this virtual server.

If a destination IP address matches the IP patterns for more than one virtual server, the longest match takes precedence. Consider the following example:

- Virtual Server 1: IP pattern 10.10.0.0, IP mask 255.255.0.0
- Virtual Server 2: IP pattern 10.10.10.0, IP mask 255.255.255.0
- Destination IP address in the packet: 10.10.10.45.
- Selected virtual server: Virtual Server 2.

The pattern associated with Virtual Server 2 matches more bits than that associated with Virtual Server 1, so IPs that match it will be sent to Virtual Server 2.

Note: Ports are also considered if a tie-breaker is required.

To configure a virtual server IP address mask by using the command line interface

At the command prompt, type:

```
add lb vserver <name>@ http -ipPattern <ipAddressPattern> -ipMask <
    ipMask> <listenPort>
<!--NeedCopy-->
```

Example:

Pattern matching based on prefix octets:

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask
    255.255.0.0 80
<!--NeedCopy-->
```

Pattern matching based on trailing octets:

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask
    0.0.255.255 80
<!--NeedCopy-->
```

Modify a pattern-based virtual server:

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
<!--NeedCopy-->
```

To configure a virtual server IP address mask by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Address Type list, select IP Pattern, and specify an IP pattern and IP mask.

Configure load balancing for commonly used protocols

September 21, 2020

In addition to Web sites and Web-based applications, other types of network-deployed applications that use other common protocols often receive large amounts of traffic and therefore benefit from load balancing. Several of these protocols require specific configurations for load balancing to work properly. Among them are FTP, DNS, SIP, and RTSP.

If you configure your NetScaler appliance to use domain names for your servers rather than IPs, you may also need to set up IP translation and masking for those servers.

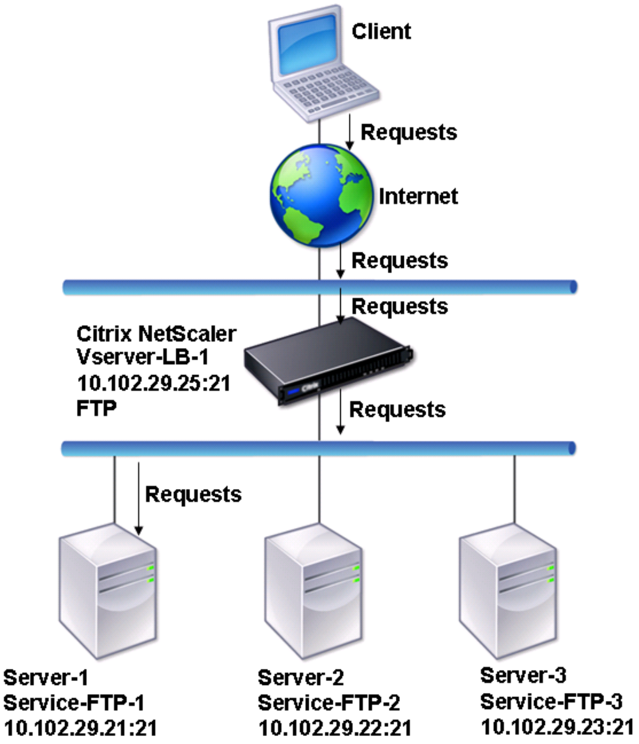
Load balance a group of FTP servers

September 21, 2020

The NetScaler appliance can be used to load balance FTP servers. FTP requires that the user initiate two connections on two different ports to the same server: the control connection, through which the client sends commands to the server, and the data connection, through which the server sends data to the client. When the client initiates an FTP session by opening a control connection to the FTP server, the appliance uses the configured load balancing method to select an FTP service, and forwards the control connection to it. The load balanced FTP server then opens a data connection to the client for information exchange.

The following diagram describes the topology of a load balancing configuration for a group of FTP servers.

Figure 1. Basic Load Balancing Topology for FTP Servers


Client
Requests
Internet
Requests
Requests
Citrix NetScaler
Vserver-LB-1
10.102.29.25:21
FTP
Requests
Requests
Server-1
Service-FTP-1
10.102.29.21:21
Server-2
Service-FTP-2
10.102.29.22:21
Server-3
Service-FTP-3
10.102.29.23:21

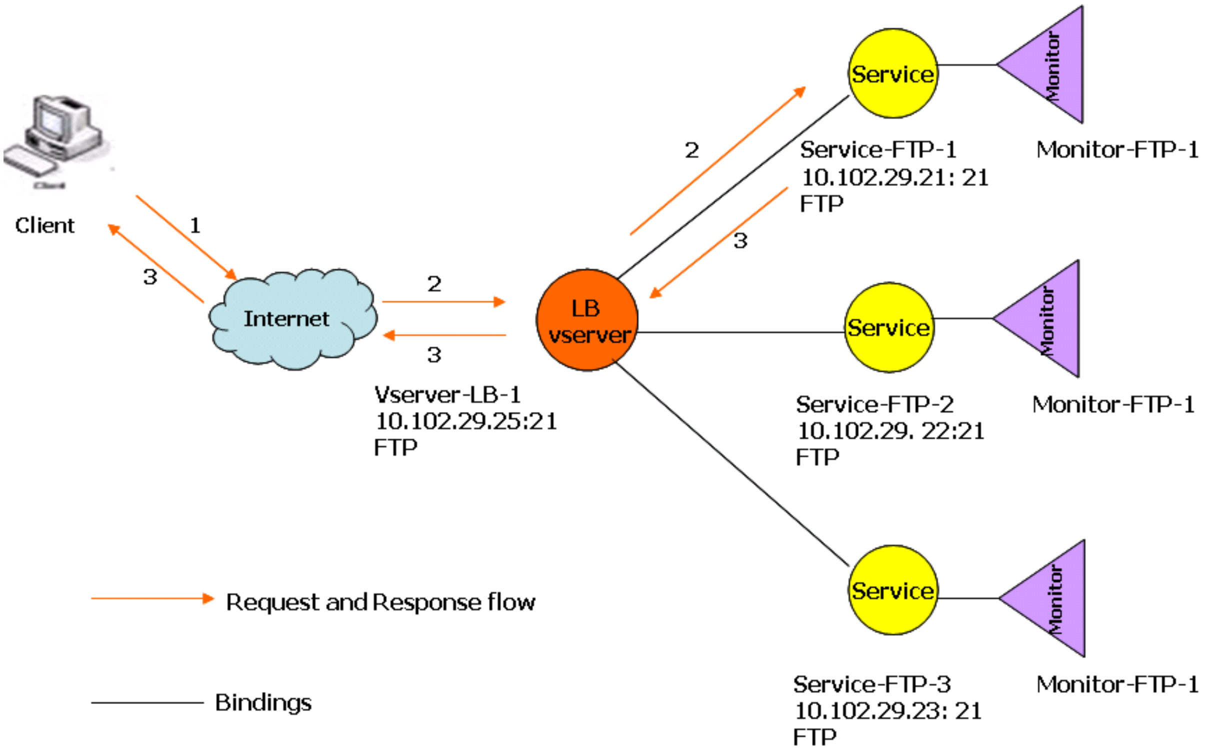
In the diagram, the services Service-FTP-1, Service-FTP-2, and Service-FTP-3 are bound to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client's connection request to one of the services using the least connection load balancing method. Subsequent requests are forwarded to the service that the appliance initially selected for load balancing.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Vserver	Vserver-LB-1	10.102.29.25	21	FTP
Services	Service-FTP-1	10.102.29.21	21	FTP
	Service-FTP-2	10.102.29.22	21	FTP
	Service-FTP-3	10.102.29.23	21	FTP
Monitors	FTP	None	None	None

The following diagram shows the load balancing entities, and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing FTP Servers Entity Model


Client
1
3
Internet
2
3
LB vserver
Vserver-LB-1
10.102.29.25:21
FTP
2
3
Service
Monitor
Service-FTP-1
10.102.29.21: 21
FTP
Monitor-FTP-1
Service
Monitor
Service-FTP-2
10.102.29. 22:21
FTP
Monitor-FTP-1
Service
Monitor
Service-FTP-3
10.102.29.23: 21
FTP
Monitor-FTP-1
Request and Response flow
Bindings

The appliance can also provide a passive FTP option to access FTP servers from outside of a firewall. When a client uses the passive FTP option and initiates a control connection to the FTP server, the FTP server also initiates a control connection to the client. It then initiates a data connection to transfer a file through the firewall.

To create services and virtual servers of type FTP, see Setting Up Basic Load Balancing. Name the entities and set the parameters to the values described in the columns of the previous table. When you configure a basic load balancing setup, a default monitor is bound to the services.

Next, bind the FTP monitor to the services by following the procedure described in the section Binding Monitors to Services.

To create FTP monitors by using the command line interface

At the command prompt, type:

```
add lb monitor <MonitorName> FTP -interval <Interval> -userName <
    UserName> -password <Password>
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password
     User
<!--NeedCopy-->
```

To create FTP monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor of type FTP, and in Special Parameters, specify a user name and password.

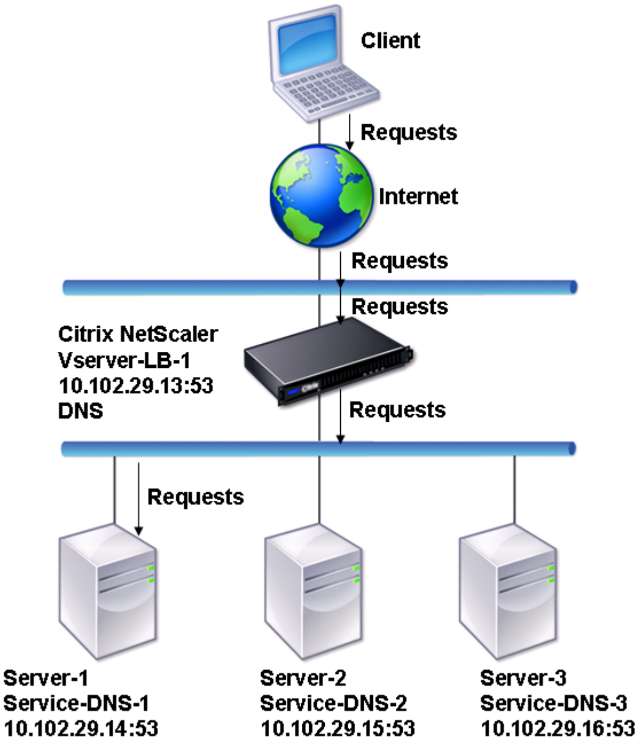
Load balance DNS servers

September 21, 2020

When you request DNS resolution of a domain name, the NetScaler appliance uses the configured load balancing method to select a DNS service. The DNS server to which the service is bound then resolves the domain name and returns the IP address as the response. The appliance can also cache DNS responses and use the cached information to respond to future requests for resolution of the same domain name. Load balancing DNS servers improves DNS response times.

The following diagram describes the topology of a load balancing configuration that load balances a group of DNS services.

Figure 1. Basic Load Balancing Topology for DNS Servers

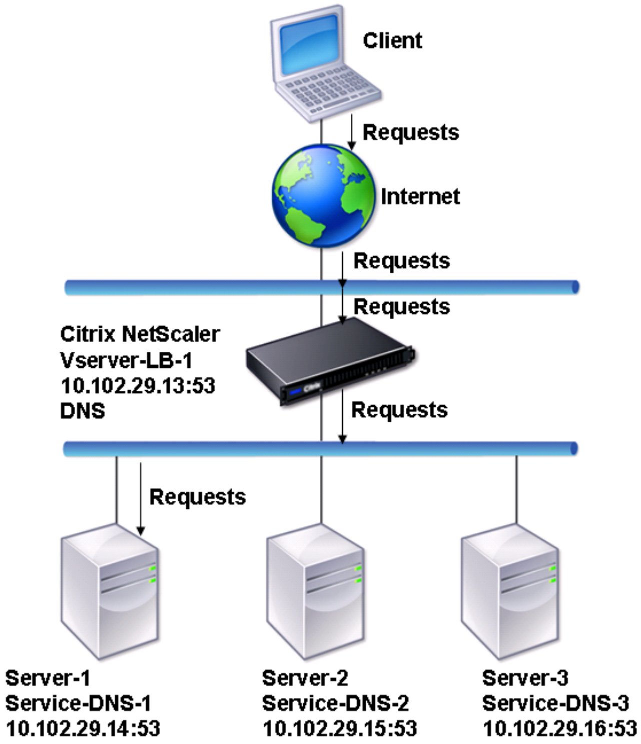

Client
Requests
Internet
Requests
Requests
Citrix NetScaler
Vserver-LB-1
10.102.29.13:53
DNS
Requests
Requests
Server-1
Service-DNS-1
10.102.29.14:53
Server-2
Service-DNS-2
10.102.29.15:53
Server-3
Service-DNS-3
10.102.29.16:53

In the diagram, the services Service-DNS-1, Service-DNS-2, and Service-DNS-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 forwards client requests to a service using the least connection load balancing method. The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.13	53	DNS
Services	Service-DNS-1	10.102.29.14	53	DNS
	Service-DNS-2	10.102.29.15	53	DNS
	Service-DNS-3	10.102.29.16	53	DNS
Monitors	monitor-DNS-1	None	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DNS Servers Entity Model


Client
Requests
Internet
Requests
Requests
Citrix NetScaler
Vserver-LB-1
10.102.29.13:53
DNS
Requests
Requests
Server-1
Service-DNS-1
10.102.29.14:53
Server-2
Service-DNS-2
10.102.29.15:53
Server-3
Service-DNS-3
10.102.29.16:53

To configure a basic DNS load balancing setup, see Setting Up Basic Load Balancing. Follow the procedures to create services and virtual servers of type DNS, naming the entities and setting the parameters using the values described in the previous table. When you configure a basic load balancing setup, the default ping monitor is bound to the services. For instructions on binding a DNS monitor to DNS services, you can also see Binding Monitors to Services.

The following procedure describes the steps to create a monitor that maps a domain name to the IP address based on a query.

To configure DNS monitors by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> DNS -query <domainName> -queryType <
    Address|ZONE> -IPAddress <ipAddress>
<!--NeedCopy-->
```

Example:

```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType
    Address -IPAddress 10.102.29.66
```

```

add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType Address -IPAddress
1000:0000:0000:0000:0005:0600:700a::888b-888d
<!--NeedCopy-->
```

To configure DNS monitors by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor of type DNS, and in Special Parameters, specify a query and query type.

Load balance domain-name based services

September 21, 2020

When you create a service for load balancing, you can provide an IP address. Alternatively, you can create a server using a domain name. The server name (domain name) can be resolved using an IPv4 or IPv6 name server, or by adding an authoritative DNS record (A record for IPv4 or AAAA record for IPv6) to the NetScaler configuration.

When you configure services with domain names instead of IP addresses, and if the name server resolves the domain name to a new IP address, the monitor bound to the service runs a health check on the new IP address, and updates the service IP address only when the IP address is found to be healthy. The monitor could be the default monitor bound to the service or you can bind any other supported monitor. It probes the service at regular intervals defined in the monitor parameters. If the domain name resolves to a new IP address, the monitor sends a fresh probe to check the health of the service. All subsequent probes are at the predefined interval.

Note: When you change the IP address of a server, the corresponding service is marked down for the first client request. The name server resolves the service IP address to the changed IP address for the next request, and the service is marked UP.

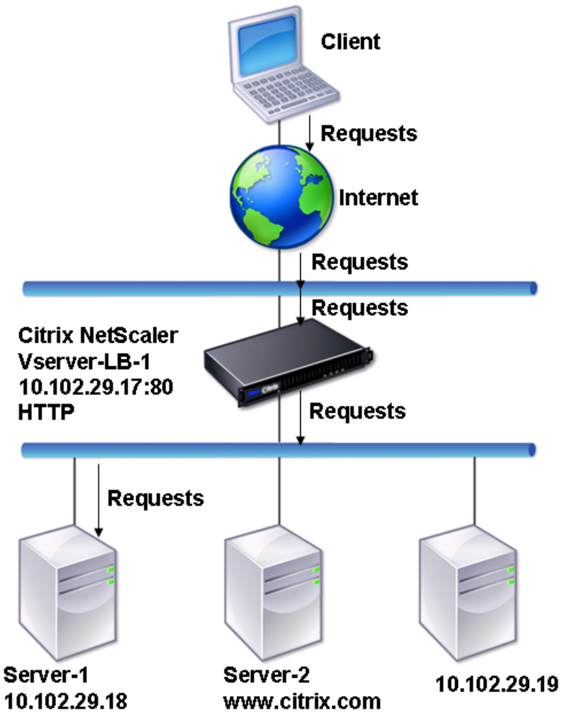
Domain-name based services have the following restrictions:

- The maximum domain name length is 255 characters.
- The Maximum Client parameter is used to configure a service that represents the domain name-based server. For example, a maxClient of 1000 is set for the services bound to a virtual server. When the connection count at the virtual server reaches 2000, the DNS resolver changes the IP address of the services. However, because the connection counter on the service is not reset, the virtual server cannot take any new connections until all the old connections are closed.
- When the IP address of the service changes, persistence is difficult to maintain.

- If the domain name resolution fails due to a timeout, the appliance uses the old information (IP address).
- When monitoring detects that a service is down, the appliance performs a DNS resolution on the service (representing the domain name-based server) to obtain a new IP address.
- Statistics are collected on a service and are not reset when the IP address changes.
- If a DNS resolution returns a code of "name error" (3), the appliance marks the service down and changes the IP address to zero.

When the appliance receives a request for a service, it selects the target service. This way, the appliance balances load on your services. The following diagram describes the topology of a load balancing configuration that load balances a group of domain-name based servers (DBS).

Figure 1. Basic Load Balancing Topology for DBS Servers


Client
Requests
Internet
Requests
Requests
Citrix NetScaler
Vserver-LB-1
10.102.29.17:80
HTTP
Requests
Requests
Server-1
10.102.29.18
Server-2
www.citrix.com
10.102.29.19

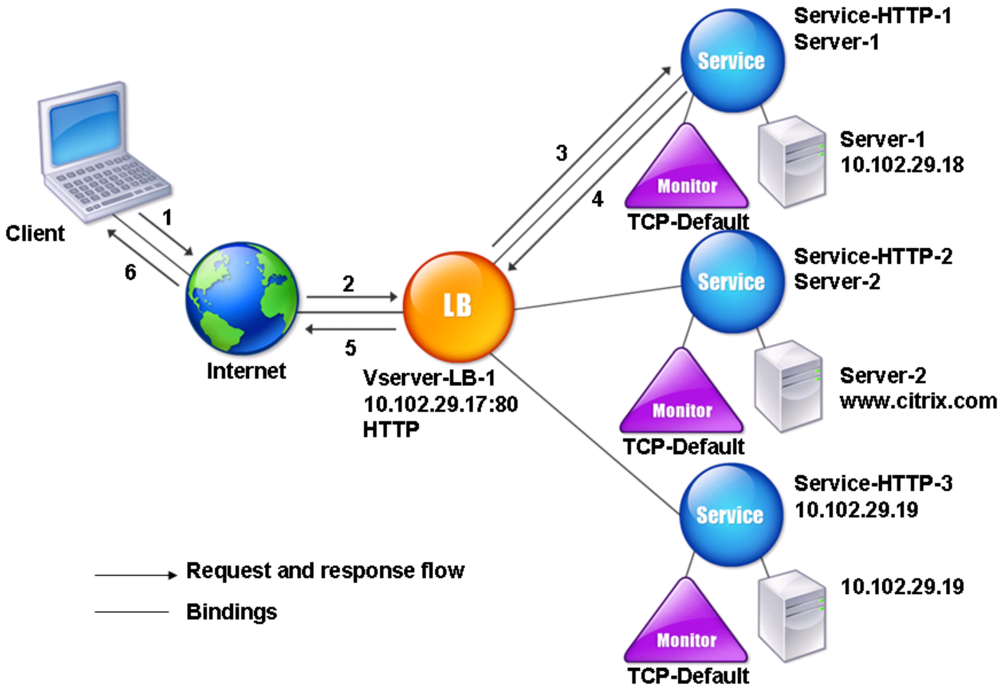
The services Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 are bound to the virtual server Vserver-LB-1. The vserver Vserver-LB-1 uses the least connection load balancing method to choose the service. The IP address of the service is resolved using the name server Vserver-LB-2.

The following table lists the names and values of the basic entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.17	80	HTTP
	Vserver-LB-2	10.102.29.20	53	DNS
Servers	server-1	10.102.29.18	80	HTTP
	server-2	www.citrix.com	80	HTTP
Services	Service-HTTP-1	server-1	80	HTTP
	Service-HTTP-2	server-2	80	HTTP
	Service-HTTP-2	10.102.29.19	80	HTTP
Monitors	Default	None	None	None
Name Server	None	10.102.29.19	None	None

The following diagram shows the load balancing entities and the values of the parameters that need to be configured on the appliance.

Figure 2. Load Balancing DBS Servers Entity Model


Service-HTTP-1
Server-1
Service
Server-1
10.102.29.18
3
Monitor
4
TCP-Default
Client
1
6
2
LB
Service-HTTP-2
Server-2
Service
Internet
5
Vserver-LB-1
10.102.29.17:80
HTTP
Server-2
www.citrix.com
Monitor
TCP-Default
Service-HTTP-3
10.102.29.19
Service
Request and response flow
Bindings
10.102.29.19
Monitor
TCP-Default

To configure a basic load balancing setup, see Setting Up Basic Load Balancing. Create the services and virtual servers of type HTTP, and name the entities and set the parameters using the values described in the previous table.

You can add, remove, enable, and disable external name servers. You can create a name server by specifying its IP address, or you can configure an existing virtual server as the name server.

To add a name server by using the command line interface

At the command prompt, type:

```
add dns nameServer <dnsVserverName>
<!--NeedCopy-->
```

Example:

```
add dns nameServer Vserver-LB-2
<!--NeedCopy-->
```

To add a name server by using the configuration utility

1. Navigate to **Traffic Management** > **DNS** > **Name Servers**.
2. Create a DNS name server of type DNS Virtual Server, and select a server from the DNS Virtual Server list.

You can also add an authoritative name server that resolves the domain name to an IP address.

> **Note**
>
> You can add a name server of type TCP, UDP or UDP_TCP to resolver DBS probes. However, if TCP and UDP name servers coexists, and a UDP name server receives a response with truncated bit, this response is not retried over TCP name server.

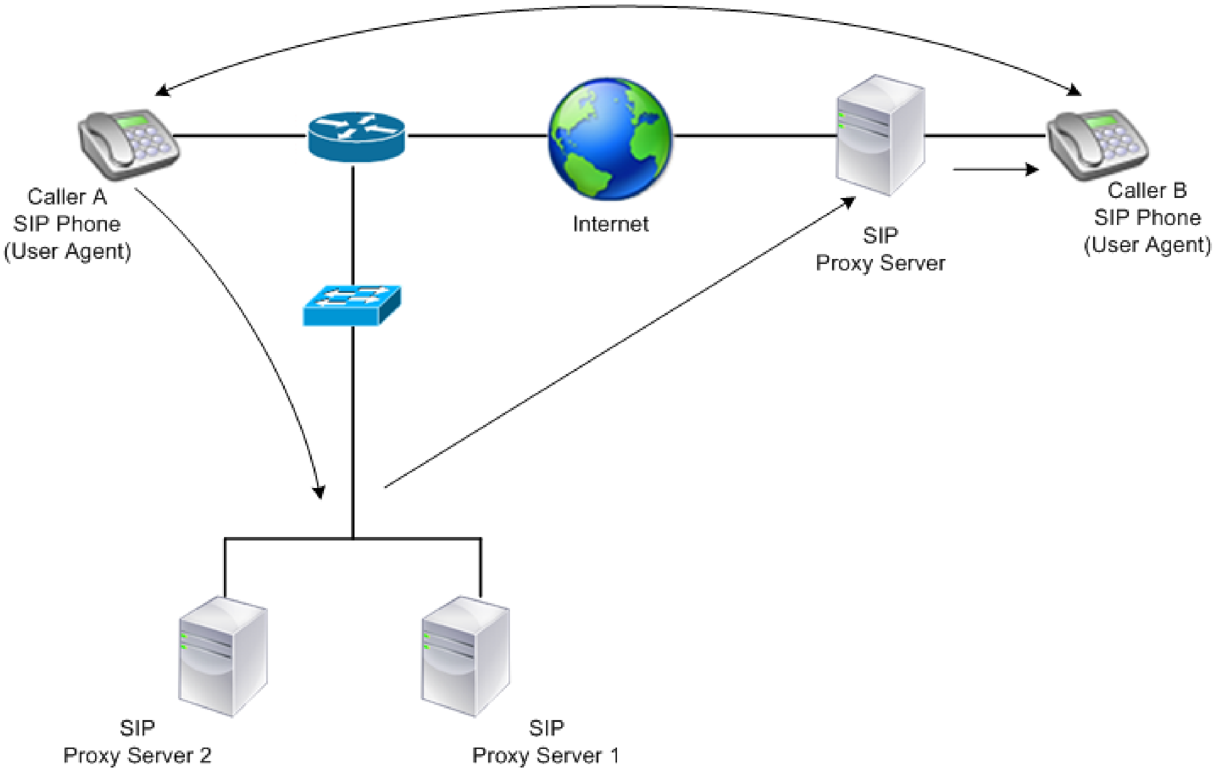
Load balance a group of SIP servers

September 21, 2020

The Session Initiation Protocol (SIP) is designed to initiate, manage, and terminate multimedia communications sessions. It has emerged as the standard for Internet telephony (VoIP). SIP messages can be transmitted over TCP or UDP. SIP messages are of two types: request messages and response messages.

The traffic in a SIP based communication system is routed through dedicated devices and applications (entities). In a multimedia communication session, these entities exchange messages. The following figure shows a basic SIP based communication system:

Figure 1. SIP Based Communication System



A NetScaler enables you to load balance SIP messages over UDP or over TCP (including TLS). You can configure the NetScaler to load balance SIP requests to a group of SIP proxy servers. To do so, you create a load balancing virtual server with the load balancing method and the type of persistence set to one of the following combinations:

- Call-ID hash load balancing method with no persistence setting
- Call-ID based persistence with least connection or round robin load balancing method
- Rule based persistence with least connection or round robin load balancing method

Also, by default, the NetScaler appends RPORT to the via header of the SIP request, so that the server sends the response back to the source IP address and port from which the request originated.

Note: For load balancing to work, you must configure the SIP proxies so that they do not add private IP addresses or private domains to the SIP header/payload. SIP proxies must add to the SIP header a domain name that resolves to the IP address of the SIP virtual server. Also, the SIP proxies must communicate with a common database to share registration information.

Server Initiated Traffic

For SIP-server initiated outbound traffic, configure RNAT on the NetScaler so that the private IP addresses used by the clients are translated into public IP addresses.

If you have configured SIP parameters that include the RNAT source or destination port, the appliance compares the values of the source and destination ports of the request packets with the RNAT source port and RNAT destination port. If one of the values matches, the appliance updates the VIA header with RPORT. The SIP response from the client then traverses the same path as the request.

For server-initiated SSL traffic, the NetScaler uses a built-in certificate-key pair. If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnatsip-127.0.0.1-5061**.

Support for Policies and Expressions

The NetScaler default expressions language contains a number of expressions that operate on Session Initiation Protocol (SIP) connections. These expressions can be bound only to SIP based (sip_udp, sip_tcp or sip_ssl) virtual servers, and to global bind points. You can use these expressions in content switching, rate limiting, responder, and rewrite policies.

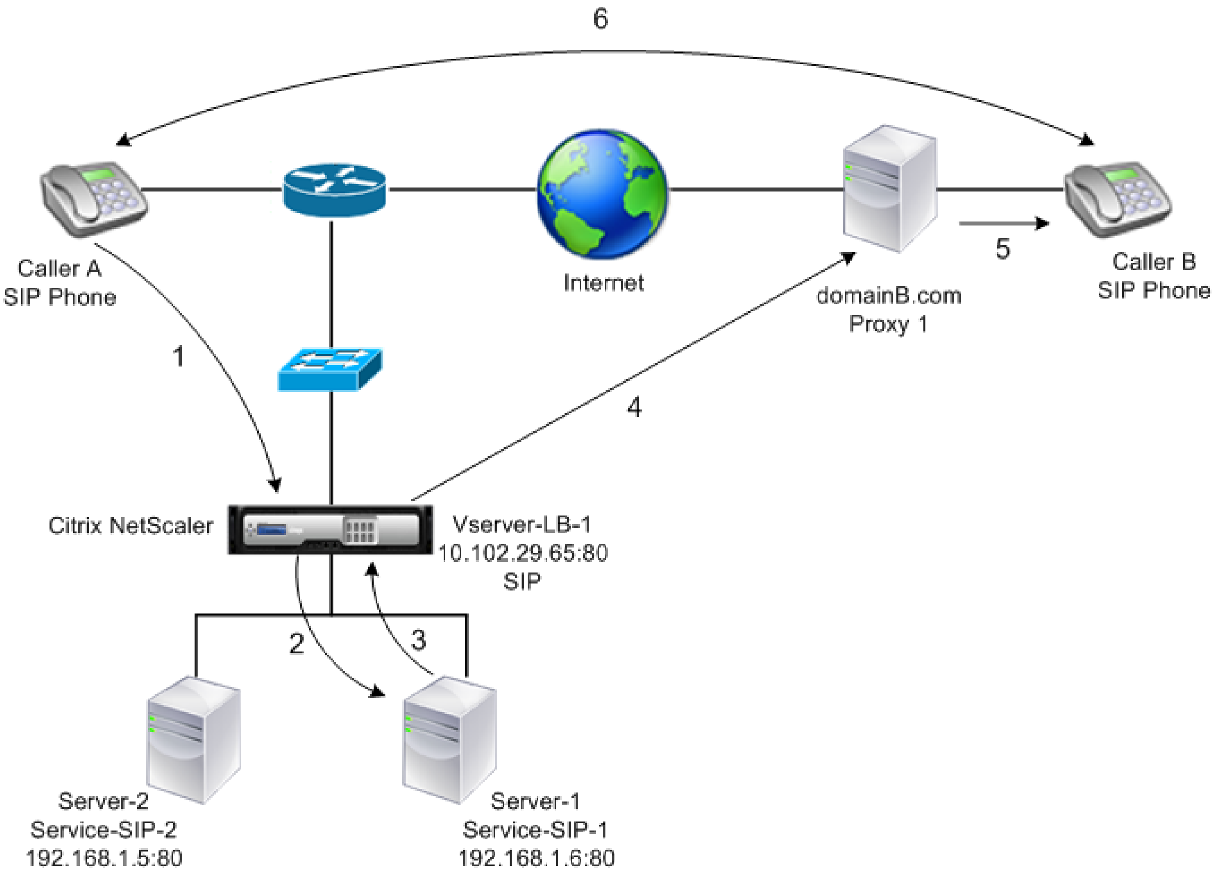
Configuring Load Balancing for SIP Signaling Traffic over TCP or UDP

The NetScaler can load balance SIP servers that send requests over UDP or TCP, including TCP traffic secured by TLS. The ADC provides the following service types to load balance the SIP servers:

- SIP_UDP – Used when SIP servers send SIP messages over UDP.
- SIP_TCP – Used when SIP servers send SIP messages over TCP.
- SIP_SSL – Used to secure SIP signaling traffic over TCP by using SSL or TLS. The NetScaler supports the following modes:
  - End-to-end TLS connection between the client, the ADC, and the SIP server.
  - TLS connection between the client and the ADC, and TCP connection between the ADC and the SIP server.
  - TCP connection between the client and the ADC, and TLS connection between the ADC and the SIP server.

The following figure shows the topology of a setup configured to load balance a group of SIP servers sending SIP messages over TCP or UDP.

Figure 2. SIP Load Balancing Topology


6
Caller A
SIP Phone
Internet
domainB.com
Proxy 1
5
Caller B
SIP Phone
1
4
Citrix NetScaler
Vserver-LB-1
10.102.29.65:80
SIP
2
3
Server-2
Service-SIP-2
192.168.1.5:80
Server-1
Service-SIP-1
192.168.1.6:80

Entity type	Name	IP address	Port	Service type / Protocol
Virtual Server	Vserver-LB-1	10.102.29.65	80	SIP_UDP / SIP_TCP / SIP_SSL
Services	Service-SIP-1	192.168.1.6	80	SIP_UDP / SIP_TCP / SIP_SSL
	Service-SIP-2	192.168.1.5	80	SIP_UDP / SIP_TCP / SIP_SSL
Monitors	Default	None	80	SIP_UDP / SIP_TCP / SIP_SSL

Following is an overview of configuring basic load balancing for SIP traffic:

1. Configure services, and configure a virtual server for each type of SIP traffic that you want to load balance:

    - **SIP_UDP** – If you are load balancing the SIP traffic over UDP.
    - **SIP_TCP** – If you are load balancing the SIP traffic over TCP.
    - **SIP_SSL** – If you are load balancing and securing the SIP traffic over TCP.

    Note: If you use SIP_SSL, be sure to create an SSL certificate-key pair. For more information, see Adding a Certificate Key Pair.

2. Bind the services to the virtual servers.
3. If you want to monitor the states of the services with a monitor other than the default (**tcp-default**), create a custom monitor and bind it to the services. The NetScaler provides two custom monitor types, **SIP-UDP** and **SIP-TCP**, for monitoring SIP services.
4. If using a SIP_SSL virtual server, bind an SSL certificate-key pair to the virtual server.
5. If you are using the NetScaler as the gateway for the SIP servers in your deployment, configure RNAT.
6. If you want to append RPORT to the SIP messages that are initiated from the SIP server, configure the SIP parameters.

To configure a basic load balancing setup for SIP traffic by using the command line interface

Create one or more services. At the command prompt, type:

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

Example:

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
<!--NeedCopy-->
```

Create as many virtual servers as necessary to handle the services that you created. The virtual server type must match the type of services that you will bind to it. At the command prompt, type:

```
add lb vserver <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
<!--NeedCopy-->
```

Bind each service to a virtual server. At the command prompt, type:

```
bind lb vserver <name> <serverName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
<!--NeedCopy-->
```

(Optional) Create a custom monitor of type SIP-UDP or SIP-TCP, and bind the monitor to the service. At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> [<interval>]

bind lb monitor <monitorName> <ServiceName>
<!--NeedCopy-->
```

Example:

```
add lb monitor mon1 sip-UDP -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

bind monitor mon1 Service-SIP-UDP-1
<!--NeedCopy-->
```

If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. At the command prompt, type: At the command prompt, type:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -CA  - skipCAName
<!--NeedCopy-->
```

Example:

```
bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1
<!--NeedCopy-->
```

Configure RNAT as required by your network topology. At the command prompt, type one of the following commands to create, respectively, an RNAT entry that uses a network address as the condition and a SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:

```
set rnat <IPAddress> <netmask>

set rnat <IPAddress> <netmask> -natip <NATIPAddress>

set rnat <aclname> [-redirectPort <port>]

set rnat <aclname> [-redirectPort <port>] -natIP <NATIPAddress>
<!--NeedCopy-->
```

Example:

```
set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50
<!--NeedCopy-->
```

If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named nsrnatsip-127.0.0.1-5061.

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]

bind ssl service <serviceName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

```
add ssl certKey c1 -cert cert.epm -key key.ky

bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
<!--NeedCopy-->
```

If you want to append RPORT to the SIP messages that the SIP server initiates, type the following command at the command prompt:

```
set lb sipParameters -rnatSrcPort <rnatSrcPort> -rnatDstPort<
    rnatDstPort> -retryDur <integer> -addRportVip <addRportVip> -
    sip503RateThreshold <sip503_rate_threshold_value>
<!--NeedCopy-->
```

Sample Configuration for load balancing the SIP traffic over UDP

```
add service service-UDP-1 10.102.29.5 SIP_UDP 80

Done

```

```
add lb vserver vserver-LB-1 SIP_UDP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-UDP-1

Done

add lb mon mon1 sip-udp -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-UDP-1

Done

set rnat 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000
     -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

Sample Configuration for load balancing the SIP traffic over TCP

```
add service service-TCP-1 10.102.29.5 SIP_TCP 80

Done

add lb vserver vserver-LB-1 SIP_TCP 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-TCP-1

Done

add lb mon mon1 sip-tcp -sipMethod REGISTER -sipuRI sip:mon@test.com -
    sipregURI sip:mon@test.com -respcode 200

```

```
Done

bind mon mon1 service-TCP-1

Done

set rnat 192.168.1.0 255.255.255.0

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000 -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

Sample Configuration for load balancing and securing SIP traffic over TCP

```
add service service-SIP-SSL-1 10.102.29.5 SIP_SSL 80

Done

add lb vserver vserver-LB-1 SIP_SSL 10.102.29.60 80

Done

bind lb vserver vserver-LB-1 service-SIP-SSL

Done

add lb mon mon1 sip-tCP -sipMethod REGISTER -sipuRI sip:mon@test.com -sipregURI sip:mon@test.com -respcode 200

Done

bind mon mon1 service-SIP-SSL

Done

bind ssl vserver Vserver-LB-1 -certkeyName CertKey-SSL-1

Done

set rnat 192.168.1.0 255.255.255.0
```

```

Done

set lb sipParameters -rnatSrcPort 5060 -rnatDstPort 5060 -retryDur 1000 -addRportVip ENABLED -sip503RateThreshold 1000

Done
<!--NeedCopy-->
```

To configure a basic load balancing setup for SIP traffic by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a virtual server of type SIP_UDP, SIP_TCP, or SIP_SSL.
2. Click the Service section, and add a service of type SIP_UDP, SIP_TCP, or SIP_SSL.
3. (Optional) Click the Monitor section, and add a monitor of type: SIP-UDP or SIP-TCP.
4. Bind the monitor to the service, and bind the service to the virtual server.
5. If you created a SIP_SSL virtual server, bind an SSL certificate key pair to the virtual server. Click the Certificates section, and bind a certificate key pair to the virtual server.
6. Configure RNAT as required by your network topology. To configure RNAT:
   a) Navigate to System > Network > Routes.
   b) On the Routes page, click the RNAT tab.
   c) In the details pane, click Configure RNAT.
   d) In the Configure RNAT dialog box, do one of the following:
      - If you want to use the network address as a condition for creating an RNAT entry, click Network and set the following parameters:
        - Network
        - Netmask
      - If you want to use an extended ACL as a condition for creating an RNAT entry, click ACL and set the following parameters:
        - ACL Name
        - Redirect Port
   e) To set a SNIP address as a NAT IP address, skip to step 7.
   f) To set a unique IP address as a NAT IP, in the Available NAT IP (s) list, select the IP address that you want to set as the NAT IP, and then click Add. The NAT IP you selected appears in the Configured NAT IP(s) list.
   g) Click Create, and then click Close.

   If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnatsip-127.0.0.1-5061**. To bind the pair:

   a) Navigate to Traffic Management > Load Balancing > Services and click the Internal Services tab.
   b) Select nsrnatsip-127.0.0.1-5061 and click **Edit**.
   c) Click the **Certificates** section and bind a certificate key pair to the internal service.

7. If you want to append RPORT to the SIP messages that the SIP server initiates, configure the SIP parameters. Navigate to Traffic Management > Load Balancing and click Change SIP settings, set the various SIP parameters.

SIP Expression and Policy Example: Compression Enabled in Client Requests

A NetScaler cannot process compressed client SIP requests, so the client SIP request fails.

You can configure a responder policy that intercepts the SIP NEGOTIATE message from the client and looks for the compression header. If the message includes a compression header, the policy responds with “400 Bad Request,” so that the client resends the request without compressing it.

At the command prompt, type the following commands to create the responder policy:

```
add responder action sipaction1 respondwith q{
 "SIP/2.0 400 Bad Requestrnrn" }


Done

add responder policy sippol1

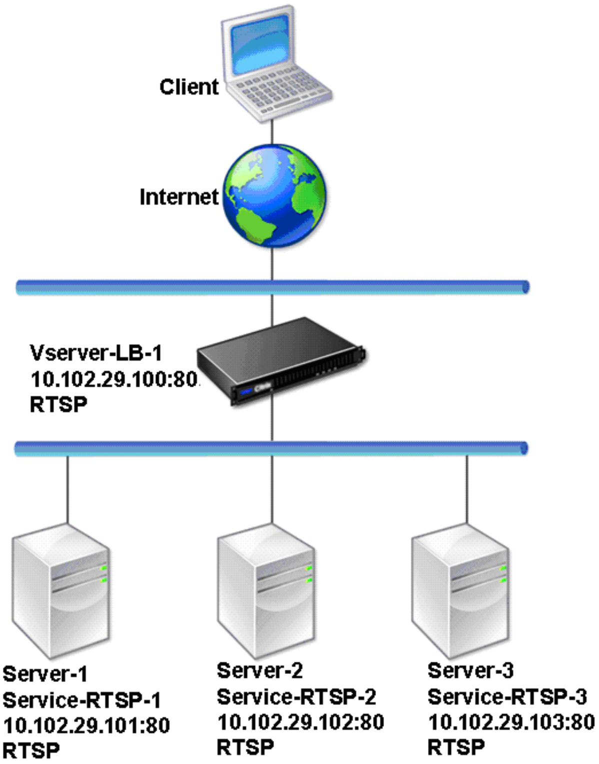
add responder policy sippol1 "SIP.REQ.METHOD.EQ("NEGOTIATE")&&SIP.REQ.HEADER("Compression").EXISTS" sipaction1
<!--NeedCopy-->
```

Load balance RTSP servers

September 21, 2020

The NetScaler appliance can balance load on RTSP servers to improve the performance of audio and video streams over networks. The following diagram describes the topology of an load balancing setup configured to load balance a group of RTSP servers.

Figure 1. Load Balancing Topology for RTSP

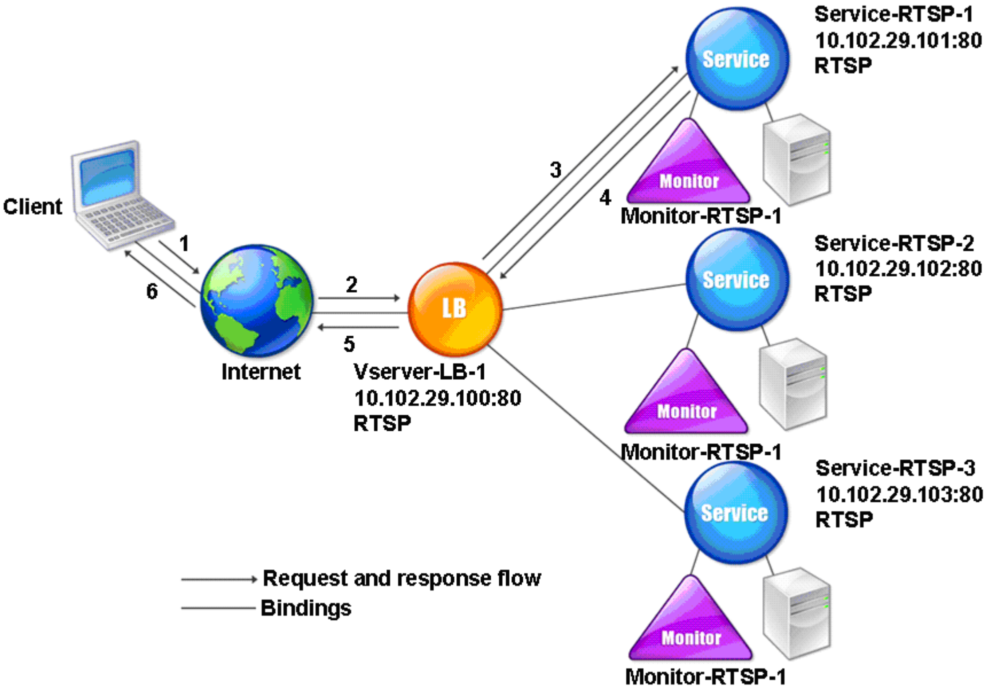

Client
Internet
Vserver-LB-1
10.102.29.100:80
RTSP
Server-1
Service-RTSP-1
10.102.29.101:80
RTSP
Server-2
Service-RTSP-2
10.102.29.102:80
RTSP
Server-3
Service-RTSP-3
10.102.29.103:80
RTSP

In the example, the services Service-RTSP-1, Service-RTSP-2, and Service-RTSP-3 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the example entities.

Entity type	Name	IP address	Port	Protocol
Virtual Server	Vserver-LB-1	10.102.29.100	554	RTSP
Services	Service-RTSP-1	10.102.29.101	554	RTSP
	Service-RTSP-2	10.102.29.102	554	RTSP
	Service-RTSP-3	10.102.29.103	554	RTSP
Monitors	Monitor-RTSP-1	None	554	RTSP

The following diagram shows the load balancing entities used in RTSP configuration.

Figure 2. Load Balancing RTSP Servers Entity Model


Service-RTSP-1
10.102.29.101:80
RTSP
Service
Monitor
Monitor-RTSP-1
Client
1
6
2
5
3
4
LB
Internet
Vserver-LB-1
10.102.29.100:80
RTSP
Service-RTSP-2
10.102.29.102:80
RTSP
Service
Monitor
Monitor-RTSP-1
Service-RTSP-3
10.102.29.103:80
RTSP
Service
Monitor
Monitor-RTSP-1
Request and response flow
Bindings

To configure a basic load balancing setup for RTSP servers, see Setting Up Basic Load Balancing. Create services and virtual servers of type RTSP. When you configure a basic load balancing setup, the default TCP-default monitor is bound to the services. To bind an RTSP monitor to these services, see Binding Monitors to Services. The following procedure describes how create a monitor that checks RTSP servers.

To configure RTSP monitors by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <type>
<!--NeedCopy-->
```

Example:

```
add lb monitor Monitor-RTSP-1 RTSP
<!--NeedCopy-->
```

To configure RTSP monitors by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type RTSP.

Load balance remote desktop protocol (RDP) servers

October 23, 2020

Remote Desktop Protocol (RDP) is a multichannel-capable protocol that allows for separate virtual channels for carrying presentation data, serial device communication, licensing information, highly encrypted data (keyboard and mouse activity), and so on.

RDP is used for providing a graphical user interface to another computer on the network. RDP is used with Windows terminal servers for providing fast access with almost real-time transmission of mouse movements and key presses even over low-bandwidth connections.

When multiple terminal servers are deployed to provide remote desktop services, the NetScaler appliance provides load balancing of the terminal servers (Windows 2003 and 2008 Server Enterprise Editions). In some cases, a user who is accessing an application remotely may want to leave the application running on the remote machine but shut down the local machine. The user therefore closes the local application without logging out of the remote application. After reconnecting to the remote machine, the user should be able to continue with the remote application. To provide this functionality, the NetScaler RDP implementation honors the routing token (cookie) set by the Terminal Services Session Directory or Broker so that the client can reconnect to the same terminal server to which it was connected previously. The Session Directory, implemented on Windows 2003 Terminal Server, is referred to as Broker on Windows 2008 Terminal Server.

When a TCP connection is established between the client and the load balancing virtual server, the NetScaler applies the specified load balancing method and forwards the request to one of the terminal servers. The terminal server checks the session directory to determine whether the client has a session running on any other terminal server in the domain.

If there is no active session on any other terminal server, the terminal server responds by serving the client request, and the NetScaler appliance forwards the response to the client.

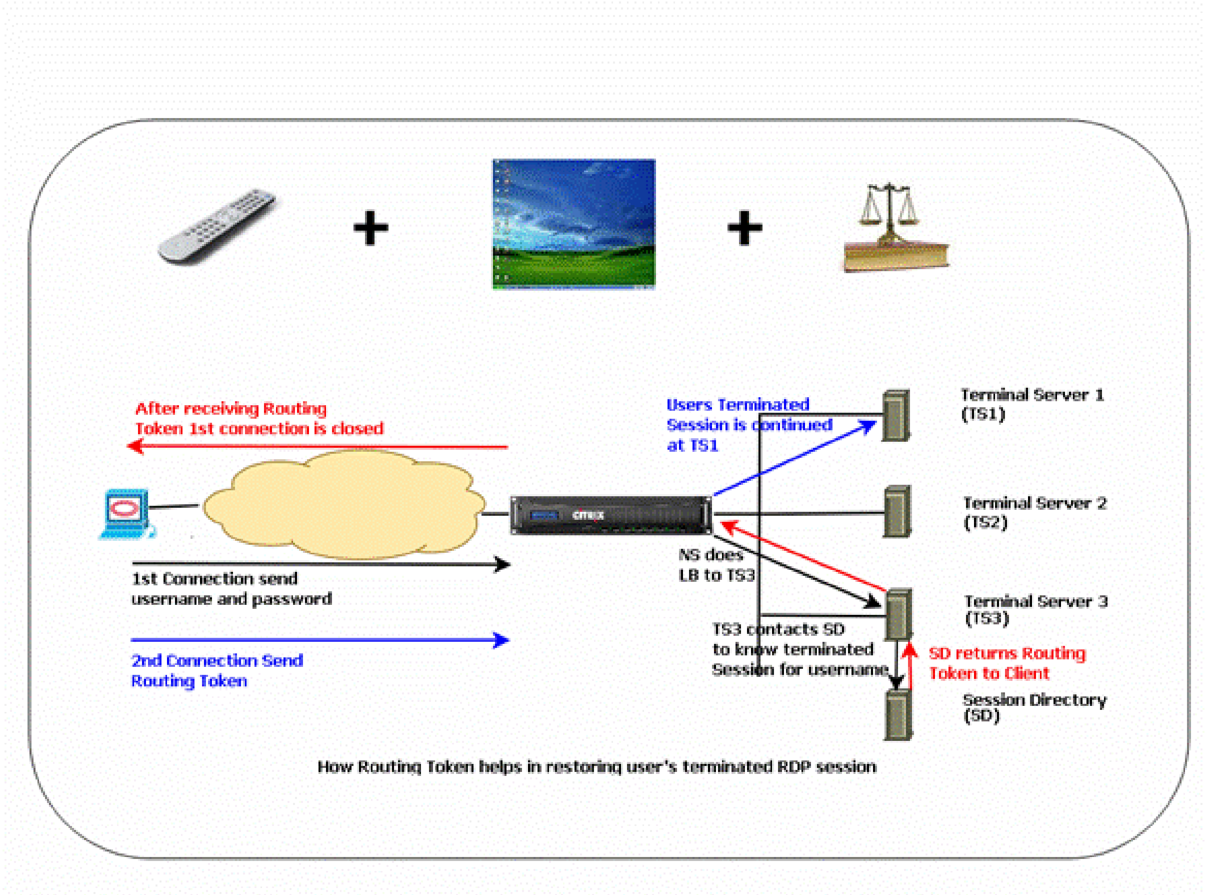
If there is an active session on any other terminal server, the terminal server that receives the request inserts a cookie (referred to as routing token) with the details of the active session and returns the packets to the NetScaler appliance, which returns the packet to the client. The server closes the connection with the client. When the client retries to connect, the NetScaler reads the cookie information and forwards the packet to the terminal server on which the client has an active session.

The user on the client machine experiences a continuation of the service and does not have to take any specific action.

Note: The Windows Session Directory feature requires the Remote Desktop client that was first released with Windows XP. If a session with a Windows 2000 or Windows NT 4.0 Terminal Server client is disconnected and the client reconnects, the server with which the connection is established is selected by the load balancing algorithm.

The following diagram describes RDP load balancing.

Figure 1. Load Balancing Topology for RDP


After receiving Routing
Token 1st connection is closed
Users Terminated
Session is continued
at TS1
Terminal Server 1
(TS1)
Terminal Server 2
(TS2)
Terminal Server 3
(TS3)
NS does
LB to TS3
1st Connection send
username and password
2nd Connection Send
Routing Token
TS3 contacts SD
to know terminated
Session for username
SD returns Routing
Token to Client
Session Directory
(SD)
How Routing Token helps in restoring user's terminated RDP session

> **Note**
>
> - When an RDP service is configured, persistence is automatically maintained by using a routing token. You need not enable persistence explicitly.
> - The NetScaler appliance supports only IP-based cookies.
> - The nsrdp.pl script is not supported on any current version of Windows servers.

Ensure that the disconnected RDP sessions are cleared on the terminal servers at the backend to avoid flapping between two terminal servers when an RDP session is disconnected without logging out. For more information, see `https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc758177(v=ws.10)##BKMK_2`

When you add an RDP service, by default, NetScaler adds a monitor of the type TCP and binds it to the service. The default monitor is a simple TCP monitor that checks whether or not a listening process exists at the 3389 port on the server specified for the RDP service. If there is a listening process at 3389, NetScaler marks this service as UP and if there is no listening process, it marks the service as DOWN.

For more efficient monitoring of an RDP service, in addition to the default monitor, you can configure a script monitor that is meant for the RDP protocol. When you configure the scripting monitor, the NetScaler opens a TCP connection to the specified server and sends an RDP packet. The monitor marks the service as UP only if it receives a confirmation of the connection from the physical server. Therefore, from the scripting monitor, the NetScaler can know whether the RDP service is ready to service a request.

The monitor is a user-type monitor and the script is located on the NetScaler at /nsconfig/monitors/nsrdp.pl. When you configure the user monitor, the NetScaler runs the script automatically. To configure the scripting monitor, add the monitor and bind it to the RDP service.

To configure RDP load balancing, create services of type RDP and bind them to an RDP virtual server.

To configure RDP load balancing services by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing setup and verify the configuration:

```
add service <name>@ <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Note: Repeat the above command to add more services.

Example

```
> add service ser1 10.102.27.182 RDP 3389
Done
> add service ser2 10.102.27.183 RDP 3389
Done
>show service ser1
ser1 (10.102. 27.182:3389) - RDP
        State: UP
…
            Server Name: 10.102.27.182
            Server ID : 0            Monitor Threshold : 0
        Down state flush: ENABLED
…
1)      Monitor Name: tcp-default
                State: UP       Weight: 1
…
                Response Time: 4.152 millisec
```

```
 Done
<!--NeedCopy-->
```

To configure RDP load balancing services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create services of type RDP.

To configure an RDP load balancing virtual server by using the command line interface

At the command prompt, type the following commands to configure an RDP load balancing virtual server and verify the configuration:

```
add lb vserver <name>@ <serviceType> <ipAddress> <port>

bind lb vserver <name>@ <serviceName>

Bind all the RDP services to be load balanced to the virtual server.
<!--NeedCopy-->
```

Example:

This example has two RDP services bound to the RDP virtual server.

```
add lb vs v1 rdP 10.102.27.186 3389
Done

bind lb vs v1 ser1
service "ser1" bound

bind lb vs v1 ser2
service "ser2" bound
Done

sh lb vs v1
v1 (10.102.27.186:3389) - RDP   Type: ADDRESS
State: UP
…
No. of Bound Services :  2 (Total)       2 (Active)
Configured Method: LEASTCONNECTION
  Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
  L2Conn: OFF

```

```
1) ser1 (10.102.27.182: 3389) - RDPState: UP   Weight: 1
2) ser2 (10.102.27.183: 3389) - RDPState: UP   Weight: 1
Done
<!--NeedCopy-->
```

To configure an RDP load balancing virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, create a virtual server of type RDP, and bind RDP services to this virtual server.

To configure a scripting monitor for RDP services by using the command line interface

At the command prompt, type the following commands:

```
add lb monitor <monitorName> USER -scriptName nsrdp.pl

bind lb monitor <monitorName> <rdpServiceName>
<!--NeedCopy-->
```

Example:

```
add service ser1 10.102.27.182 RDP 3389

add lb monitor RDP_MON USER -scriptName nsrdp.pl

bind lb monitor RDP_MON ser1

<!--NeedCopy-->
```

To configure a scripting monitor for RDP services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type USER.
2. In Special Parameters, in the Script Name list, select nsrdp.pl, and then bind this monitor to an RDP service.

Use case 1: SMPP load balancing

September 21, 2020

Millions of short messages are exchanged daily between individuals and value-added service providers, such as banks, advertisers, and directory services, by using the short message peer to peer (SMPP) protocol. Often, message delivery is delayed because servers are overloaded and traffic is not optimally distributed among the servers. The NetScaler supports SMPP load balancing and provides optimal distribution of messages across your servers, preventing poor performance and outages.

The NetScaler performs load balancing on the server side when messages are received from clients and on the client side when messages are received from servers.

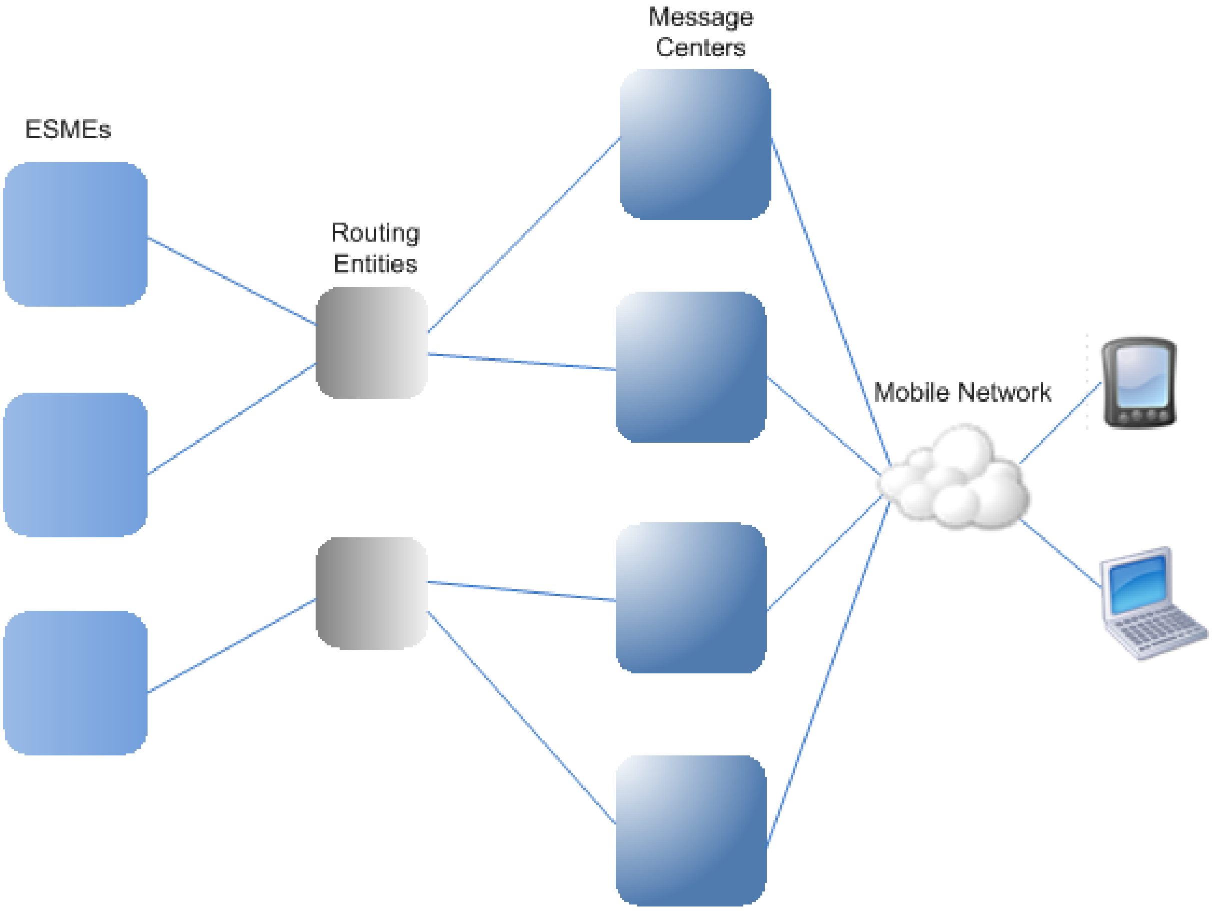
Load balancing of SMPP messages by the NetScaler provides the following benefits:

- Better load distribution on servers, which translates to faster response time to end users
- Server health monitoring and better failover capabilities
- Quick and easy addition of new servers (message centers) without changing the client configuration
- High availability

Introduction to SMPP

SMPP is an application layer protocol for transfer of short messages between External Short Message Entities (ESME), Routing Entities (RE) and Message Centers (MC) over long-lived TCP connections. It is used for sending short message service (SMS) messages between friends, contacts, and third parties such as banks (mobile banking), advertisers (mobile commerce), and directory services. Messages from an ESME (non-mobile entity) arrive at the MC, which distributes them to short message entities (SMEs) such as mobile phones. SMPP is also used by SMEs to send short messages to third parties (for example, for purchase of products, bill payment, and funds transfer). These messages arrive at the MC and are forwarded to the destination MC or ESME.

The following diagram shows the different SMPP entities: ESMEs, REs, and MCs, in a mobile network.



Architecture Overview of the Different SMPP Entities in a Mobile Network

Note: The terms client and ESME are used interchangeably throughout the document.

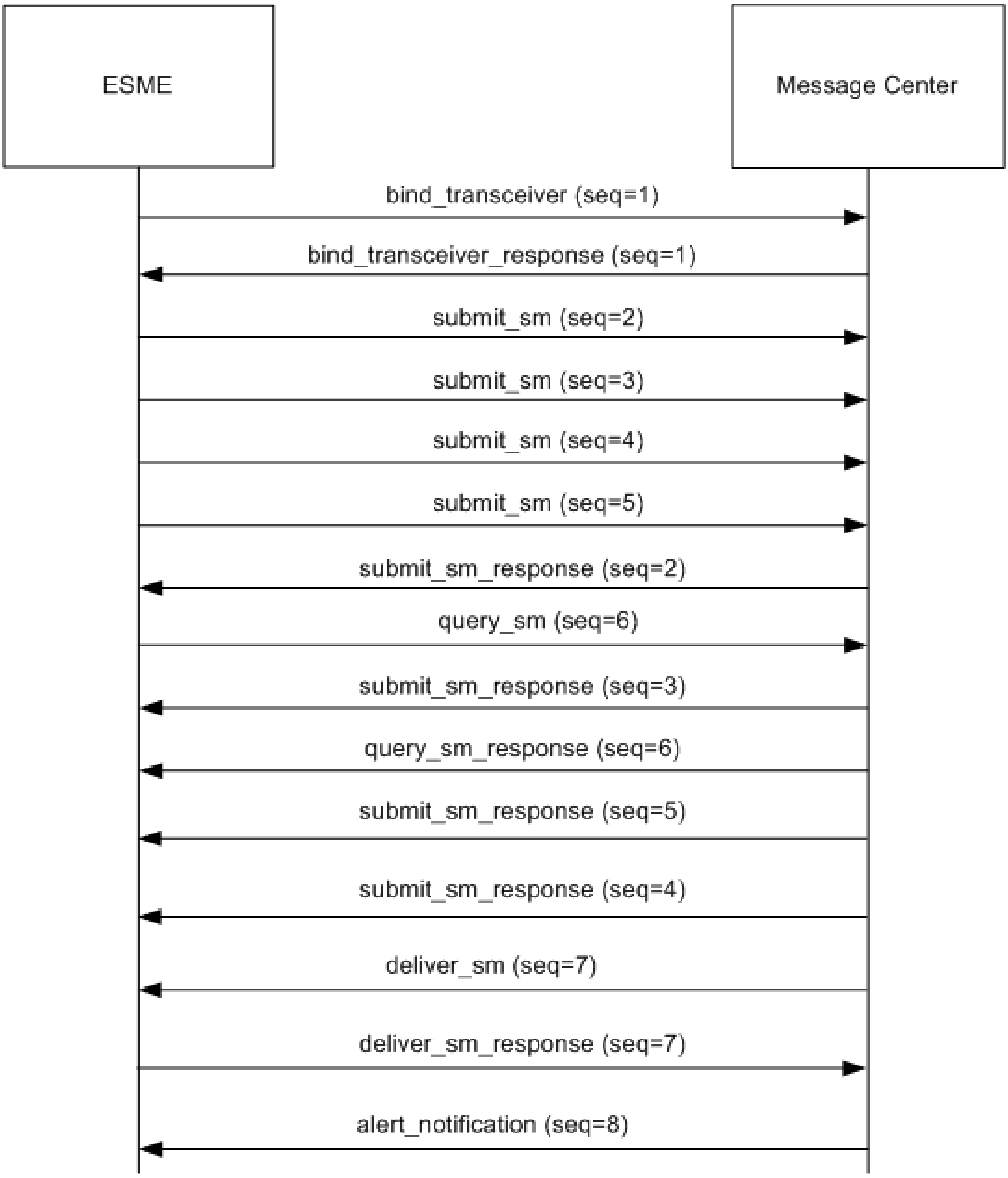
An ESME (client) opens a connection to the MC in one of the three modes: as a transmitter, a receiver, or a transceiver. As a transmitter, it can only submit messages for delivery. As a receiver, it can only receive messages. As a transceiver, the ESME can both submit and receives messages. The ESME sends the MC one of the three messages (also known as PDUs): bind_transmitter, bind_receiver, or bind_transceiver. The MC responds with a bind_transmitter_resp, bind_receiver_resp, or bind_transceiver_resp, as appropriate for the request.

After the connection is established, the ESME can, depending on the mode in which it is bound to the MC, send a submit_sm or data_sm message, receive a deliver_sm or data_sm message, or send and receive any of these types of messages. The ESME can also send ancillary messages, such as query_sm, replace_sm, and cancel_sm, to query the status of an earlier message delivery, replace an earlier message with a new message, or cancel an undelivered message.

If a message is not delivered because an ESME is not available or a mobile subscriber is not online, the message is queued. Later, when the MC detects that the mobile subscriber is now reachable, it sends an alert_notification PDU to the ESME over a receiver or transceiver session, requesting delivery of any queued messages.

Each request PDU has a unique sequence number. The response PDU has the same sequence number as the original request. Because message exchange over SMPP can be in asynchronous mode, an ESME or an MC can send multiple requests at a time. The sequence number plays a crucial role in returning the response in the same SMPP session. In other words, the sequence number makes request and response matching possible.

The following diagram shows how the traffic flow uses the various PDUs when the ESME binds as a transceiver.


ESME
Message Center
bind_transceiver (seq=1)
bind_transceiver_response (seq=1)
submit_sm (seq=2)
submit_sm (seq=3)
submit_sm (seq=4)
submit_sm (seq=5)
submit_sm_response (seq=2)
query_sm (seq=6)
submit_sm_response (seq=3)
query_sm_response (seq=6)
submit_sm_response (seq=5)
submit_sm_response (seq=4)
deliver_sm (seq=7)
deliver_sm_response (seq=7)
alert_notification (seq=8)

Limitation:

The NetScaler appliance does not support outbind operations. That is, a message center cannot initiate an SMPP session with an ESME through the NetScaler appliance.

How SMPP Load Balancing Works on the NetScaler

An ESME (client) sends a bind message to open a connection to the NetScaler. The ADC authenticates each ESME and, if successful, responds with an appropriate message. The NetScaler establishes a connection with each message center and load balances all the messages among these message centers. When the ADC receives a message from a client, it reuses an open connection to the message center or sends a bind request to a message center if an open connection is not available.

The ADC can load balance messages originating from the clients and from the servers. It can monitor the health of the message centers and handle concatenated messages. It also provides content switching support for the message centers.

Messages Originating from the ESMEs

Each ESME must be added as a user on the NetScaler for authentication. The client establishes a TCP connection with an SMPP virtual server configured on the ADC by sending a bind request. The ADC authenticates the client and, if successful, parses the bind message. The ADC then sends the request to the message center selected by the configured load balancing method. If a connection to the message center is not available for reuse, the ADC opens a TCP connection with the message center by sending a new bind request to the message center.

Before forwarding the response (submit_sm_resp or data_sm_resp) from the message center to the client, the ADC adds a custom server ID to the message ID to identify the message center for ancillary operations, such as query, replace, or cancel requests for a message, by the client. Requests from other clients are load balanced in the same way.

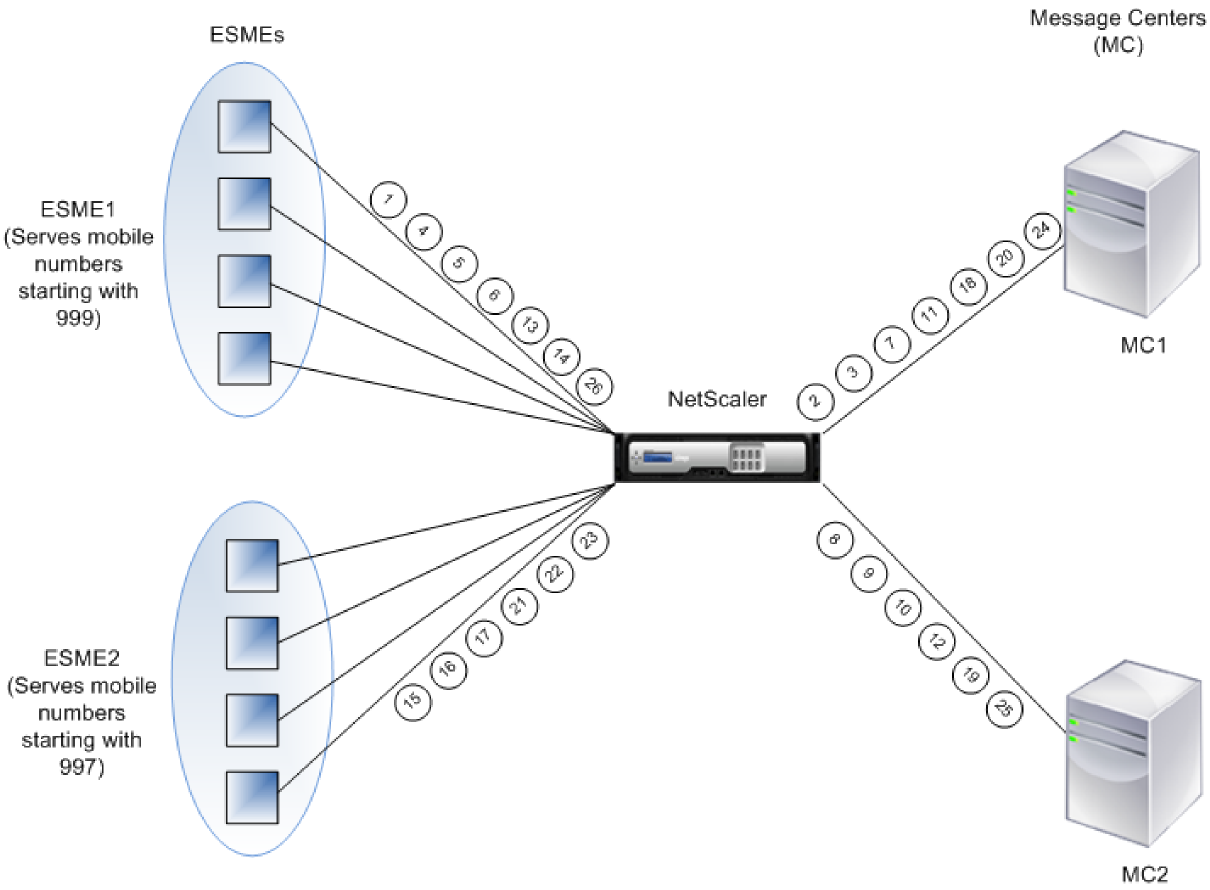
In the original bind request, a client specifies the address range that it can serve. This range is used for forwarding deliver_sm or data_sm messages from the message centers to the clients.

Messages Originating from a Message Center

ESMEs that can handle a specific address range are grouped into a cluster. All the nodes in a cluster provide the same credentials. Within a cluster, only the round robin method is used for load balancing. To deliver mobile originated (MO) messages, the message center sends a deliver_sm message to the NetScaler. If a cluster that can serve the destination address range (for example, numbers starting with 998) is bound to the ADC, it selects that cluster, and then load balances the message among the ESME nodes in that cluster.

If an ESME that can serve deliver_sm messages for the address range is not bound to the ADC, and message queuing is enabled, the message is queued until such a client binds to the ADC in a receiver or transceiver mode. You can specify the size of the queue.

The following diagram illustrates the internal flow of PDUs between ESMEs, NetScaler, and the message centers. For simplicity, only two ESMEs and two message centers are shown.


ESMEs
Message Centers (MC)
ESME1 (Serves mobile numbers starting with 999)
ESME2 (Serves mobile numbers starting with 997)
NetScaler
MC1
MC2
1
4
5
6
13
14
26
2
3
7
11
18
20
24
15
16
17
21
22
23
8
9
10
12
19
25

Flow of messages (PDUs):

1. ESME1 sends bind request to NetScaler
2. NetScaler sends bind request to MC1
3. MC1 sends bind response to NetScaler
4. NetScaler sends bind response to ESME1
5. ESME1 sends submit_sm(1) to NetScaler
6. ESME1 sends submit_sm(2) to NetScaler
7. NetScaler forwards submit_sm(1) to MC1
8. NetScaler sends bind request to MC2
9. MC2 sends bind response to NetScaler
10. NetScaler forwards submit_sm(2) to MC2
11. MC1 sends submit_sm_resp(1) to NetScaler
12. MC2 sends submit_sm_resp(2) to NetScaler
13. NetScaler forwards submit_sm_resp(1) to ESME1
14. NetScaler forwards submit_sm_resp(2) to ESME1
15. ESME2 sends bind request to NetScaler
16. NetScaler sends bind response to ESME2
17. ESME2 sends submit_sm(3) to NetScaler

18. NetScaler forwards submit_sm(3) to MC1
19. MC2 sends deliver_sm to NetScaler (ESME2 serves the address range specified in the message)
20. MC1 sends submit_sm_resp(3) to NetScaler
21. NetScaler forwards submit_sm_resp(3) to ESME2
22. NetScaler forwards deliver_sm to ESME2
23. ESME2 sends deliver_sm_resp to NetScaler
24. MC1 sends alert_notification to NetScaler (ESME1 serves the address range specified in the message)
25. NetScaler forwards deliver_sm_resp to MC2
26. NetScaler forwards the alert_notification to ESME1

Health Monitoring of Message Centers

By default, a TCP_default monitor is bound to an SMPP service, but you can bind a custom monitor of type SMPP. The custom monitor opens a TCP connection to the message center and sends an enquire_link packet. Depending on the success or failure of the probe, the service is marked UP or DOWN.

Content Switching on Message Centers

Message centers can accept multiple connections (or bind requests) from ESMEs. You can configure the NetScaler to content switch these requests on the basis of the SMPP bind parameters. Following are some common expressions for configuring methods to select a message center:

- Based on the address range: In the following sample expression, the ADC selects a specific message center if the address range starts at 988.

Example:

SMPP.BINDINFO.ADDRESS_RANGE.CONTAINS(“^988”)

- Based the ESME ID: In the following sample expression, the ADC selects a specific message center if the ESME ID equals ESME1.

Example:

SMPP.BINDINFO.SYSTEM_ID.EQ(“ESME1”)

- Based on the ESME type: In the following sample expression, the ADC selects a specific message center if the ESME type is VMS. VMS stands for voice mail system.

Example:

SMPP.BINDINFO.SYSTEM_TYPE.EQ(“VMS”)

- Based on the type of number (TON) of the ESME: In the following sample expression, the ADC selects a specific message center if TON equals 1 (1 stands for an international number.)

Example:

SMPP.BINDINFO.ADDR_TON.EQ(1)

- Based on the number plan indicator (NPI) of the ESME: In the following sample expression, the ADC selects a specific message center if NPI equals 0 (0 stands for an unknown connection.)

Example:

SMPP.BINDINFO.ADDR_NPI.EQ(0)

- Based on the bind type: In the following sample expression, the ADC selects a specific message center if the bind type is TRANSCEIVER. (A transceiver can send and receive messages.)

Example:

SMPP.BINDINFO.TYPE.EQ(TRANSCEIVER)

Concatenated Message Handling

An SMS can hold a maximum of 140 bytes. Longer messages must be broken down into smaller parts. If the destination mobile is capable, the messages are combined and delivered as one long SMS. The NetScaler forwards the fragments of a message to the same message center. Each message contains a reference number, a sequence number, and the total number of fragments. The reference number is the same for each fragment of a long message. The sequence number specifies that position of the particular fragment in the complete message. After all the fragments are received, the ESME combines the fragments into one long message and delivers the message to the mobile subscriber.

If a client disconnects from an active connection, the connection to the message center is not closed. It is reused for requests from other clients.

Limitation

Message IDs, from the message center, longer than 59 bytes are not supported. If the message ID length returned by the message center is more than 59 bytes, ancillary operations fail and the NetScaler responds with an error message.

Configuring SMPP Load Balancing on the NetScaler

Perform the following tasks to configure SMPP load balancing on the ADC:

1. Add an SMPP user. The ADC authenticates the user before it accepts a bind request from the user. The user is typically an ESME.
2. Add a load balancing virtual server, specifying the protocol as SMPP.
3. Add a service, specifying the protocol as SMPP, and a custom server ID that is unique for each server. Bind the service to the load balancing virtual server created earlier.

4. Optionally, create a service group and add services to the service group.
5. Optionally, add a monitor of type SMPP-ECV and bind it to the service. A TCP-default monitor is bound by default.
6. Set the SMPP parameters, such as client mode and message queue.

To configure SMPP load balancing by using the command line

At the command prompt, type:

```
add smpp user <username> -password <password>
add service <name> <IP> SMPP <port>  - customserverID <customserverID>
add lb vserver <name> <IP> SMPP <port>
bind lb vserver <name> <service name>
set smpp param
<!--NeedCopy-->
```

Example

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmethod ENCMTHD_2
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmethod ENCMTHD_2
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CustomServerID ab -CKA NO -TCPB NO -CMP NO
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED -usip NO -useproxyport YES -sp ON -cltTimeout 180 -svrTimeout 360 -CustomServerID xy -CKA NO -TCPB NO -CMP NO
add lb vserver smppvs SMPP 10.102.239.179 2775 -persistenceType NONE -cltTimeout 180
bind lb vserver smppvs smmpsvc2
bind lb vserver smppvs smmpsvc
set smpp param -addrrange "d*"
<!--NeedCopy-->
```

To configure SMPP load balancing by using the configuration utility

1. Navigate to System > User Administration > SMPP Users, and add an SMPP user.
2. Navigate to Traffic Management > Load Balancing > Configure SMPP Parameters, and set the parameters as required by your deployment.
3. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a virtual server of type SMPP.

4. Click in the Service section, add a service of type SMPP, and specify a Server ID.

Use case 2: Configure rule based persistence based on a name-value pair in a TCP byte stream

September 21, 2020

Some protocols transmit name-value pairs in a TCP byte stream. The protocol in the TCP byte stream in this example is the Financial Information eXchange (FIX) protocol. In its traditional, non-XML implementation, the FIX protocol enables two hosts communicating over a network to exchange business or trade-related information as a list of name-value pairs (called "FIX fields"). The field format is `<tag>=<value><delimiter>`. This traditional tag-value format makes the FIX protocol ideal for the use case.

The tag in a FIX field is a numeric identifier that indicates the meaning of the field. For example, the tag 35 indicates the message type. The value after the equal sign holds a specific meaning for the given tag and is associated with a data type. For example, a value of A for the tag 35 indicates that the message is a logon message. The delimiter is the nonprinting "Start of Header" (SOH) ASCII character (0x01), which is the caret symbol (^). Each field is also assigned a name. For example, the field with tag 35 is the msgType field. Following is an example of a logon message.

8=FIX.4.1 9=61 35=A 49=INVMGR 56=BRKR 34=1 52=20000426-12:05:06 98=0 108=30 10=157

Your choice of persistence type for a tag-value list such as the one shown above is determined by the options that are available to you for extracting a particular string from the list. Token-based persistence methods require you to specify the offset and length of the token that you want to extract from the payload. The FIX protocol does not allow you to do that, because the offset of a given field and the length of its value can vary from one message to another (depending on the message type, the preceding fields, and the lengths of the preceding values) and from one implementation to another (depending on whether custom fields have been defined). Such variations make it impossible to predict the exact offset of a given field or to specify the length of the value that is to be extracted as the token. In this case, therefore, rule based persistence is the preferred persistence type.

Assume that a virtual server fixlb1 is load balancing TCP connections to a farm of servers hosting instances of a FIX-enabled application, and that you want to configure persistence for connections on the basis of the value of the SenderCompID field, which identifies the firm sending the message. The tag for this FIX field is 49 (shown in the earlier logon message example).

To configure rule based persistence for the load balancing virtual server, set the persistence type for the load balancing virtual server to RULE and configure the rule parameter with an expression. The expression must be one that extracts the portion of the TCP payload in which you expect to find the

SenderCompID field, typecasts the resulting string to a name-value list based on the delimiters, and then extracts the value of the SenderCompID field (tag 49), as follows:

set lb vserver fixlb1 -persistenceType RULE -rule “CLIENT.TCP.PAYLOAD(300).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“\4

Note: Backslash characters have been used in the expression because this is a CLI command. If you are using the configuration utility, do not enter the backslash characters.

If the client sends a FIX message that contains the name-value list in the earlier logon message example, the expression extracts the value INVMGR, and the NetScaler appliance creates a persistence session based on this value.

The argument to the PAYLOAD() function can be as large as you deem is necessary to include the SenderCompID field in the string extracted by the function. Optionally, you can use the SET_TEXT_MODE(IGNORECASE) function if you want the appliance to ignore case when extracting the value of the field, and the HASH function to create a persistence session based on a hash of the extracted value. The following expression uses the SET_TEXT_MODE(IGNORECASE) and HASH functions:

CLIENT.TCP.PAYLOAD(500).TYPECAST_NVLIST_T(‘=’,’^’).SET_TEXT_MODE(IGNORECASE).VALUE(“49”).HASH

Following are more examples of rules that you can use to configure persistence for FIX connections (replace
<tag> with the tag of the field whose value you want to extract):

- To extract the value of any FIX field in the first 300 bytes of the TCP payload, you can use the expression CLIENT.TCP.PAYLOAD(300).BEFORE_STR(“^”).AFTER_STR(“<tag>=”).
- To extract a string that is 20 bytes long at offset 80, cast the string to a name-value list, and then extract the value of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).SUBSTR(80,20).TYPE
  tag>”).
- To extract the first 100 bytes of the TCP payload, cast the string to a name-value list, and extract the value of the third occurrence of the field that you want, use the expression CLIENT.TCP.PAYLOAD(100).TYPECAST_NVLIST_T(‘=’,’^’).VALUE(“<tag>“,2).
  Note: If the second argument that is passed to the
  VALUE() function is
  n, the appliance extracts the value of the
  (n+1)
  <sup>th</sup> instance of the field because the count starts from zero (
  0).

Following are more examples of rules that you can use to configure persistence. Only the payload-based expressions can evaluate data being transmitted through the FIX protocol. The other expressions are more general expressions for configuring persistence based on lower networking protocols.

- CLIENT.TCP.PAYLOAD(100)

- CLIENT.TCP.PAYLOAD(100).HASH
- CLIENT.TCP.PAYLOAD(100).SUBSTR(5,10)
- CLIENT.TCP.SRCPORT
- CLIENT.TCP.DSTPORT
- CLIENT.IP.SRC
- CLIENT.IP.DST
- CLIENT.IP.SRC.GET4
- CLIENT.IP.DST.GET4
- CLIENT.ETHER.SRCMAC.GET6
- CLIENT.ETHER.DSTMAC.GET5
- CLIENT.VLAN.ID

Use case 3: Configure load balancing in direct server return mode

September 21, 2020

Load balancing in direct server return (DSR) mode allows the server to respond to clients directly by using a return path that does not flow through the NetScaler appliance. In DSR mode, however, the appliance can continue to perform health checks on services. In a high-data volume environment, sending server traffic directly to the client in DSR mode increases the overall packet handling capacity of the appliance because the packets do not flow through the appliance.

DSR mode has the following features and limitations:

- It supports one-arm mode and inline mode.
- The appliance ages out sessions based on idle timeout.
- Because the appliance does not proxy TCP connections (that is it does not send SYN-ACK to the client), it does not completely shut out SYN attacks. By using the SYN packet rate filter, you can control the rate of SYNs to the server. To control the rate of SYNs, set a threshold for the rate of SYNs. To get protection from SYN attacks, you must configure the appliance to proxy TCP connections. However, that requires the reverse traffic to flow through the appliance.
- In a DSR configuration, the NetScaler appliance does not replace the load balancing virtual server's IP address with the destination server's IP address. Instead, it forwards packets to a service by using the server's MAC address. The VIP must be configured on the server and ARP must be disabled for the VIP which is configured on the server to prevent the client request from bypassing the appliance when it is configured in one-arm mode. For example, a user needs to configure VIP in the loopback interface and disable the ARP for the same VIP.
- The appliance obtains the server's MAC address from the monitor bound to the service. However, custom user monitors (monitors of type USER), which use scripts stored on the NetScaler appliance, do not learn a server's MAC address. If you use only custom monitors in a DSR con-

figuration, for each request the virtual server receives, the appliance attempts to resolve the destination IP address to a MAC address (by sending ARP requests). Because the destination IP address is a virtual IP address owned by the NetScaler appliance, the ARP requests always resolve to the MAC address of the NetScaler interface. Consequently, all traffic received by the virtual server is looped back to the appliance. If you use user monitors in a DSR configuration, you must also configure another monitor of a different type (for example, a PING monitor) for the services, ideally with a longer interval between probes, so that the MAC address of the servers can be learned.

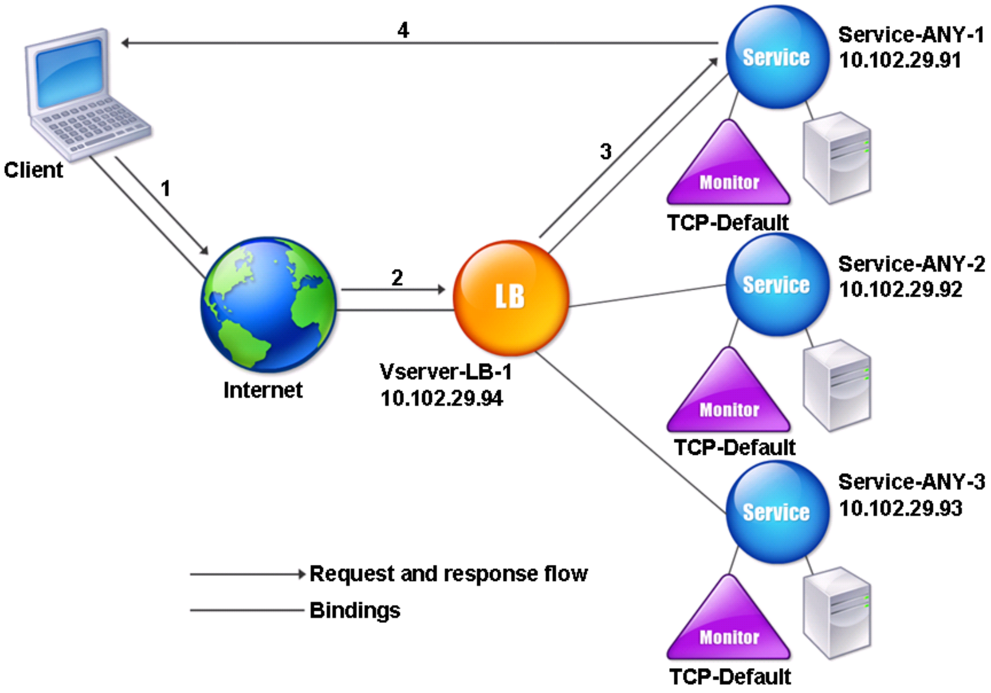
- The Citrix ADC appliance learns the server L2 parameters from the monitor that is bound to the service. For UDP-ECV monitors, configure a receive string to enable the appliance to learn the L2 parameters of the server. If the receive string is not configured and the server does not respond, then the appliance does not learn the L2 parameters but the service is set to UP. The traffic for this service is blackholed.

In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service, and the service responds to clients directly, bypassing the NetScaler appliance. The following table lists the names and values of the entities configured on the NetScaler appliance in DSR mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 1. Entity Model for Load Balancing in DSR Model


4
Service
Service-ANY-1
10.102.29.91
Client
1
3
Monitor
TCP-Default
Service
Service-ANY-2
10.102.29.92
2
LB
Internet
Vserver-LB-1
10.102.29.94
Monitor
TCP-Default
Service
Service-ANY-3
10.102.29.93
Request and response flow
Bindings
Monitor
TCP-Default

For the appliance to function correctly in DSR mode, the destination IP in the client request must be unchanged. Instead, the appliance changes the destination MAC to that of the selected server. This setting enables the server to determine the client MAC address for forwarding requests to the client while bypassing the server.

Next, you configure a basic load balancing setup as described in Setting Up Basic Load Balancing, naming the entities and setting the parameters using the values described in the previous table.

After you configure the basic load balancing setup, you must customize it for DSR mode. To do this, you configure a supported load balancing method, such as the Source IP Hash method with a sessionless virtual server. You also need to set the redirection mode to allow the server to determine the client MAC address for forwarding responses and bypass the appliance.

After you configure the load balancing method and redirection mode, you need to enable the USIP mode on each service. The service then uses the source IP address when forwarding responses.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

Example

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless
    enabled
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure the load balancing method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server, select Redirection Mode as MAC Based, and method as SOURCEIPHASH.
3. In Traffic Settings, select Sessionless Load Balancing.

To configure a service to use source IP address by using the command line interface

At the command prompt, type:

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

To configure a service to use source IP address by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and in Traffic Settings, select Use Source IP Address.

Some additional steps are required in certain situations, which are described in the succeeding sections.

Use case 4: Configure LINUX servers in DSR mode

September 21, 2020

The LINUX operating system requires that you set up a loopback interface with the NetScaler appliance virtual IP address (VIP) on each load balanced server in the DSR cluster.

To configure LINUX server in DSR mode

To create a loop back interface with the NetScaler appliance's VIP on each load balanced server, at the Linux OS prompt type the following commands:

```
ifconfig dummy0 up

ifconfig dummy0:0 inet <netscaler vip> netmask 255.255.255.255 up

echo 1 > /proc/sys/net/ipv4/conf/dummy0/arp_ignore

echo 2 > /proc/sys/net/ipv4/conf/dummy0/arp_announce
<!--NeedCopy-->
```

Then, run the software that re-maps the TOS id to VIP.

Note: Add the correct mappings to the software before running it. In the preceding commands, the LINUX server uses dummy0 to connect to the network. When you use this command, type the name of the interface that your LINUX server uses to connect to the network.

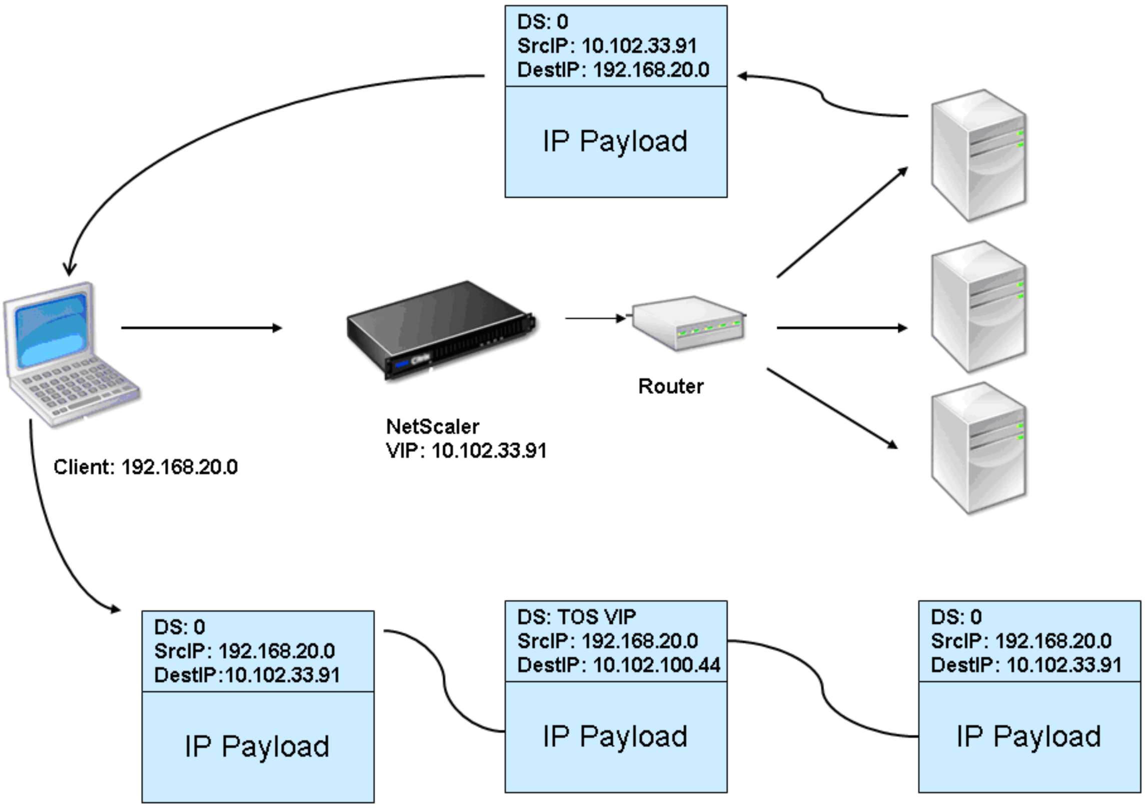
Use case 5: Configure DSR mode when using TOS

September 21, 2020

Differentiated services (DS), also known as TOS (Type of Service), is a field that is part of the IPv4 packet header. The equivalent field in the IPv6 header is Traffic Class. TOS is used by upper layer protocols for optimizing the path for a packet. The TOS information encodes the NetScaler appliance virtual IP address (VIP), and the load balanced servers extract the VIP from it.

In the following scenario, the appliance adds the VIP to the **TOS** field in the packet and then forwards the packet to the load balanced server. The load balanced server then responds directly to the client, bypassing the appliance, as illustrated in the following diagram.

Figure 1. The NetScaler appliance in DSR mode with TOS


DS: 0
SrcIP: 10.102.33.91
DestIP: 192.168.20.0
IP Payload
Router
NetScaler
VIP: 10.102.33.91
Client: 192.168.20.0
DS: 0
SrcIP: 192.168.20.0
DestIP:10.102.33.91
IP Payload
DS: TOS VIP
SrcIP: 192.168.20.0
DestIP: 10.102.100.44
IP Payload
DS: 0
SrcIP: 192.168.20.0
DestIP: 10.102.33.91
IP Payload

The TOS feature is customized for a controlled environment, as described below:

- The environment must not have any stateful devices, such as stateful firewall and TCP gateways, in the path between the appliance and the load balanced servers.
- Routers at all the entry points to the network must remove the TOS field from all incoming packets to make sure that the load balanced server does not confuse another TOS field with that added by the appliance.
- Each server can have only 63 VIPs.
- The intermediate router must not send out ICMP error messages regarding fragmentation. The client will not understand the message, as the source IP address will be the IP address of the load balanced server and not the NetScaler VIP.
- TOS is valid only for IP-based services. You cannot use domain name based services with TOS.

In the example, Service-ANY-1 is created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to the service, and the service responds to clients directly, bypassing the appliance. The following table lists the names and values of the entities configured on the appliance in DSR mode.

Entity Type	Name	IP Address	Protocol
Virtual server	Vserver-LB-1	10.102.33.91	ANY
Services	Service-ANY-1	10.102.100.44	ANY
Monitors	PING	None	None

DSR with TOS requires that load balancing be set up on layer 3. To configure a basic load balancing setup for Layer 3, see Setting Up Basic Load Balancing. Name the entities and set the parameters using the values described in the previous table.

After you configure the load balancing setup, you must customize the load balancing setup for DSR mode by configuring the redirection mode to allow the server to decapsulate the data packet and then respond directly to the client and bypass the appliance.

After specifying the redirection mode, you can optionally enable the appliance to transparently monitor the server. This enables the appliance to transparently monitor the load balanced servers.

To configure the redirection mode for the virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -m <Value> -tosId <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
<!--NeedCopy-->
```

To configure the redirection mode for the virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and in Redirect Mode, select TOS ID.

To configure the transparent monitor for TOS by using the command line interface

At the command prompt, type:

```
add monitor <MonitorName> <Type> -destip <DestinationIP> -tos <Value> -
    tosId <Value>
<!--NeedCopy-->
```

Example:

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes  -tosId 3
<!--NeedCopy-->
```

To create the transparent monitor for TOS by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Create a monitor, select TOS, and type the TOS ID that you specified for the virtual server.

Wildcard TOS Monitors

In a load balancing configuration in DSR mode using TOS field, monitoring its services requires a TOS monitor to be created and bound to these services. A separate TOS monitor is required for each load balancing configuration in DSR mode using TOS field, because a TOS monitor requires the VIP address and the TOS ID to create an encoded value of the VIP address. The monitor creates probe packets in which the **TOS** field is set to the encoded value of the VIP address. It then sends the probe packets to the servers represented by the services of a load balancing configuration.

With many load balancing configurations, creating a separate custom TOS monitor for each configuration is a significant, cumbersome task. Managing these TOS monitors is also a significant task. Now, you can create wildcard TOS monitors. You need to create only one wildcard TOS monitor for all load balancing configurations that use the same protocol (for example, TCP or UDP).

A wildcard TOS monitor has the following mandatory settings:

- Type = `<protocol>`
- TOS = Yes

The following parameters can be set to a value or can be left blank:

- Destination IP
- Destination Port
- TOS ID

A wildcard TOS monitor (with destination IP, Destination port, and TOS ID not set) bound to a DSR service automatically learns the TOS ID and the VIP address of the load balancing virtual server. The monitor creates probe packets with TOS field set to the encoded VIP address and then sends the probe packets to the server represented by the DSR service.

To create a wildcard TOS monitor by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <Type> -tos YES

show lb monitor <monitorName>
<!--NeedCopy-->
```

To bind a wildcard TOS monitor to a service by using the CLI

At the command prompt, type:

```
bind lb monitor <monitorName> <serviceName>

show lb monitor <monitorName>
<!--NeedCopy-->
```

To create a wildcard TOS monitor by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitors**.
2. Add a monitor with the following parameter settings:
   - Type = `<protocol>`
   - TOS = YES

To bind a wildcard TOS monitor to a service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Open a service and bind a wildcard TOS monitor to it.

In the following sample configuration, V1, V2, and V3 are load balancing virtual servers of type ANY and has TOS ID set to 1, 2, and 3 respectively. S1, S2, S3, S4, and S5 are services of type ANY. S1 and S2 are bound to both V1 and V2. S3, S4, and S5 and bound to both V1 and V3. WLCD-TOS-MON is a wildcard TOS monitor with type TCP and is bound to S1, S2, S3, S4, and S5.

WLCD-TOS-MON automatically learns the TOD ID and VIP address of virtual servers bound to S1, S2, S3, S4, and S5.

Because S1 is bound to V1 and V2, WLCD-TOS-MON creates two types of probe packets for S1, one with TOS field set to the encoded VIP address (203.0.113.1) of V1 and the other with the VIP address (203.0.113.2) of V2. The NetScaler then sends these probe packets to the server represented by S1. Similarly, WLCD-TOS-MON creates probe packets for S2, S3, S4, and S5.

```
add lb monitor WLCD-TOS-MON  TCP -tos YES

Done

add lb vserver V1 ANY 203.0.113.1 * -m TOS  - tosID 1

Done

add lb vserver V2 ANY 203.0.113.2 * -m TOS  - tosID 2

Done

add lb vserver V3 ANY 203.0.113.3 * -m TOS  - tosID 3

Done

add service S1 198.51.100.1 ANY *

Done

add service S2 198.51.100.2 ANY *

Done

add service S3 198.51.100.3 ANY *

Done

add service S4 198.51.100.4 ANY *

Done

add service S5 198.51.100.5 ANY *

Done

bind lb monitor WLCD-TOS-MON S1

Done

bind lb monitor WLCD-TOS-MON S2

Done

```

```
bind lb monitor WLCD-TOS-MON S3

Done

bind lb monitor WLCD-TOS-MON S4

Done

bind lb monitor WLCD-TOS-MON S5

Done

bind lb vserver V1 S1, S2, S3, S4, S5

Done

bind lb vserver V2, S1, S2

Done

bind lb vserver V3 S3, S4, S5

Done
<!--NeedCopy-->
```

Use case 6: Configure load balancing in DSR mode for IPv6 networks by using the TOS field

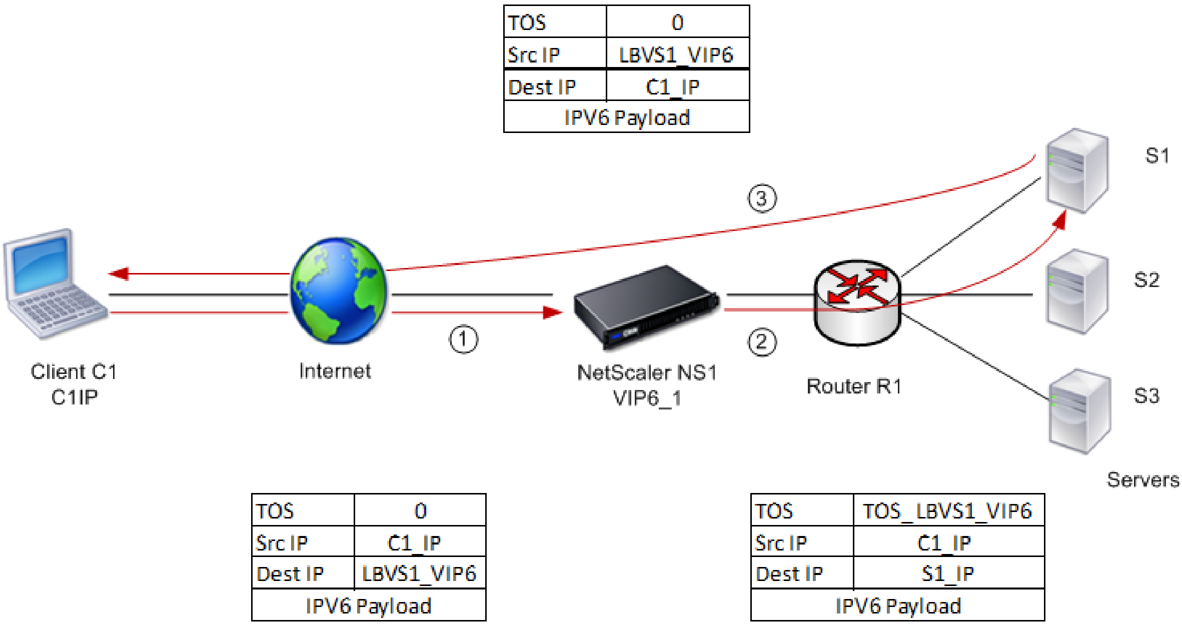
September 21, 2020

You can configure load balancing in Direct Server Return (DSR) mode for IPv6 networks by using the Type of Service (TOS) field when the NetScaler appliance and the servers are in different networks.

Note: The TOS field is also called the Traffic Class field.

In DSR mode, when a client sends a request to a VIP6 address on a NetScaler appliance, the appliance forwards this request to the server by changing the destination IPv6 address of the packet to the IPv6 address of the server and sets an encoded value of the VIP6 address in the TOS (also called traffic class) field of the IPv6 header. You can configure the server to use the information in the TOS field to derive the VIP6 address from the encoded value, which is then used as source IP address in response packets. Response traffic directly goes to the client, bypassing the appliance.

Consider an example where a load balancing virtual server LBVS1, configured on a NetScaler appliance NS1, is used to load balance traffic across servers S1, S2, and S3. The NetScaler appliance NS1 and the servers S1, S2, and S3 are in different networks so router R1 is deployed between NS1 and the servers.


TOS 0
Src IP LBVS1_VIP6
Dest IP C1_IP
IPV6 Payload
③
S1
S2
S3
Servers
Client C1
C1IP
Internet
①
NetScaler NS1
VIP6_1
②
Router R1
TOS 0
Src IP C1_IP
Dest IP LBVS1_VIP6
IPV6 Payload
TOS TOS_LBVS1_VIP6
Src IP C1_IP
Dest IP S1_IP
IPV6 Payload

The following table lists the settings used in this example.

Entities	Name
IPv6 address of client C1	C1_IP (for reference purposes only)
Load balancing virtual server on NS1	LBVS1
IPv6 address of LBVS1	LBVS1_VIP6 (for references purpose only)
TOS value	TOS_LBVS1_VIP6 (for references purpose only)
Service for server S1 on NS1	SVC_S1
IPv6 address for server S1	S1_IP (for references purpose only)
Service for server S2 on NS1	SVC_S2
IPv6 address for server S1	S2_IP (for references purpose only)
Service for server S3 on NS1	SVC_S3
IPv6 address for server S1	S3_IP (for references purpose only)

Following is the traffic flow in the example scenario:

1. Client C1 sends a request to virtual server LBVS1.

2. LBVS1's load balancing algorithm selects server S1 and the appliance opens a connection to S1. NS1 sends the request to S1 with:
    - TOS field set to TOS_LBVS1_VIP6.
    - Source IP address as C1_IP.
3. The server S1, on receiving the request, uses the information in the TOS field to derive the LBVS1_VIP6 address, which is the IP address of the virtual server LBVS1 on NS1. The server directly sends the response to C1, bypassing the appliance, with:
    - Source IP address set to the derivedLBVS1_VIP6 address so that the client communicates to the virtual server LBVS1 on NS1 and not to server S1.

To configure load balancing in DSR Mode using TOS, perform the following steps on the appliance

1. Enable USIP mode globally.
2. Add the servers as services.
3. Configure a load balancing virtual server with a TOS value.
4. Bind the services to the virtual server.

To configure load balancing in DSR Mode using TOS by using the command line interface

At the command prompt, type:

```
enable ns mode USIP

add service <serviceName> <IP> <serviceType> <port>
<!--NeedCopy-->
```

Repeat the above command as many times as necessary to add each server as a service on the NetScaler appliance.

```
add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -
    tosId <positive_integer>

bind lb vserver <vserverName> <serviceName>
<!--NeedCopy-->
```

To enable USIP mode by using the configuration utility

Navigate to System > Settings > Configure Modes, and select Use Source IP Address.

To create services by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create a load balancing virtual server and bind services by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server.
2. Click in the Service section to bind a service to this virtual server.

Use case 7: Configure load balancing in DSR mode by using IP Over IP

January 22, 2021

You can configure your NetScaler appliance to use direct server return (DSR) mode across Layer 3 networks by using IP tunneling, also called *IP over IP* configuration. As with standard load balancing configurations for DSR mode, this allows servers to respond to clients directly instead of using a return path through the NetScaler appliance, improving response times and throughput. As with standard DSR mode, the NetScaler appliance monitors the servers and performs health checks on the application ports.

With IP over IP configuration, the NetScaler appliance and the servers do not need to be on the same Layer 2 subnet. Instead, the NetScaler appliance encapsulates the packets before sending them to the destination server. After the destination server receives the packets, it decapsulates the packets, and then sends its responses directly to the client.

To configure IP over IP DSR mode on your NetScaler appliance, you must do the following:

- Create a load balancing virtual server. Set the protocol to ANY and set the mode to IPTUNNEL.
- Create services. Create a service for each of your back-end applications. Bind the services that you created to the virtual server.
- Configure for decapsulation. You can configure either a NetScaler appliance or a back-end server to act as a decapsulator.

Configure a load balancing virtual server

Configure a virtual server to handle requests to your applications. Assign a service type of ANY and set the forwarding method to IPTUNNEL. Optionally, configure the virtual server to operate in sessionless mode. You can configure any load balancing method that you want to use.

To create and configure a load balancing virtual server for IP over IP DSR by using the command line interface

At the command prompt type the following command to configure a load balancing virtual server for IP over IP DSR and verify the configuration:

```
add lb vserver <name> serviceType <serviceType> IPAddress <ip> Port <port> -lbMethod <method> -m <ipTunnelTag> -sessionless <sessionless>

show lb vserver <name>
<!--NeedCopy-->
```

Example:

In the following example, we have selected the load balancing method as sourceIPhash and configured sessionless load balancing.

```
add lb vserver Vserver-LB-1 ANY 10.102.29.60 * -lbMethod SourceIPHash -m IPTUNNEL -sessionless enabled
<!--NeedCopy-->
```

To create and configure a load balancing virtual server for IP over IP DSR by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Create a virtual server, and specify Redirection Mode as **IP Tunnel Based**.

Configure services for IP over IP DSR

After creating your load-balanced server, You must configure one service for each of your applications. The service handles traffic from the NetScaler appliance to those applications, and allows the NetScaler appliance to monitor the health of each application.

You assign a service type of ANY and configure it for USIP mode. Optionally, you can also bind a monitor of type IPTUNNEL to the service for tunnel-based monitoring.

To create and configure a service for IP over IP DSR by using the command line interface

At the command prompt, type the following commands to create a service and optionally, create a monitor and bind it to the service:

```
add service <serviceName> <serverName> <serviceType> <port> -usip <usip>

add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <iptunnel>

bind service <serviceName> -monitorName <monitorName>
<!--NeedCopy-->
```

Example:

In the following example, we are creating a monitor of type IPTUNNEL:

```
add monitor mon-1 PING -destip 10.102.29.60 -iptunnel yes
add service Service-DSR-1 10.102.30.5 ANY * -usip yes
bind service Service-DSR-1 -monitorName mon-1
<!--NeedCopy-->
```

To configure a monitor by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Monitors**.
2. Create a monitor, and select **IP Tunnel**.

To create and configure a service for IP over IP DSR by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Services**.
2. Create a service and, in Settings, select Use Source IP Address.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt type the following command:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-DSR-1
<!--NeedCopy-->
```

To bind a service to a load balancing virtual server by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and click in the Services section to bind a service to the virtual server.

Decapsulator configuration

- When a NetScaler appliance is used as a decapsulator, an IP tunnel must be created in the NetScaler appliance. For details, see Configuring IP Tunnels.

Example configuration:

```
    add lb vserver v1 any 1.1.1.1 * -m IPTUNNEL

    add service s1 2.2.2.2 ANY *

    bind lb vserver v1 s1

    add iptunnel tun1 <snip_in_encap> netmask *

    add ns ip 1.1.1.1 255.255.255.255  - type vip  - arp disabled

    add lb vserver v1 any 1.1.1.1 *

    add service s1 <actualserverip> ANY *

    bind lb vserver v1 s1
<!--NeedCopy-->
```

- When a back-end server is used as a decapsulator, the back-end configuration varies depending on the server type. The steps involved in configuring a back-end server as a decapsulator are;

1. Configure a loop back interface.
2. Add a route through tunnel interface.

Note: Make sure that the tunnel modules are installed in the system.

Example configuration:

In this example, 1.1.1.1 is the NetScaler virtual IP (VIP) address and 2.2.2.2 is the back-end server IP address.

The VIP address is configured in the loopback interface and a route is added through the tunnel interface. The `modprobe ipip` command is used for enabling the tunnel interface.

```
    add lb vserver v1 ANY 1.1.1.1 80 -m IPTUNNEL

    add service svc1 2.2.2.2 ANY 80 -usip YES -useproxyport NO

    bind lb vserver v1 svc1

    ifconfig lo inet 1.1.1.1 netmask 255.255.255.255

    modprobe ipip

    echo 1 > /proc/sys/net/ipv4/conf/tunl0/arp_ignore

    echo 2 > /proc/sys/net/ipv4/conf/tunl0/arp_announce

    ifconfig tunl0 1.1.1.1 netmask 255.255.255.255 up

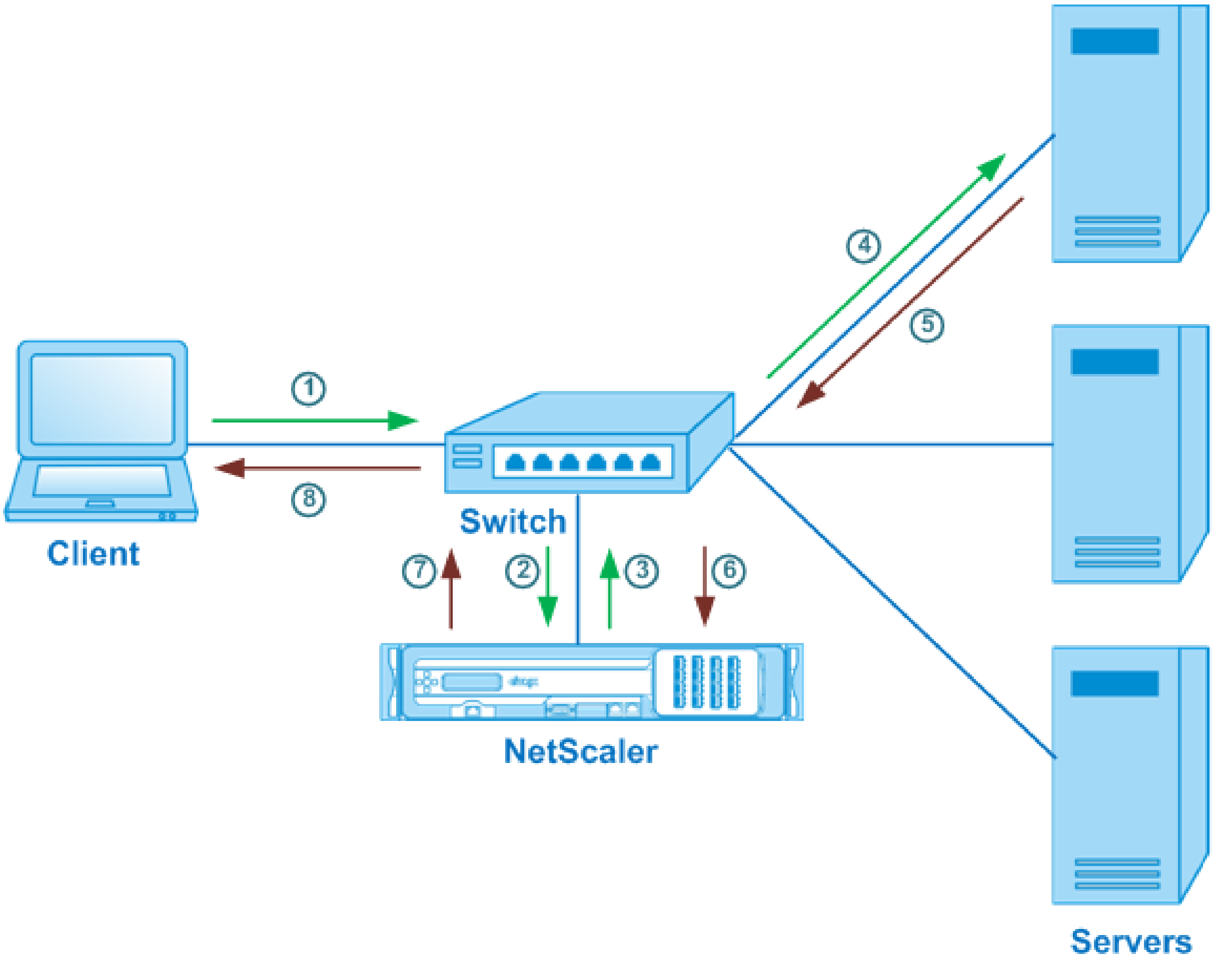
    route add -host 1.1.1.1 dev tunl0
<!--NeedCopy-->
```

Use case 8: Configure load balancing in one-arm mode

September 21, 2020

In a one-arm setup, you connect the NetScaler appliance to the network through a single VLAN. The appliance receives the request from the client on a single VLAN and it sends the request to the server on the same VLAN. This is one of the simplest deployment scenarios, where the router, the servers and the appliance are all connected to the same switch. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service, as is shown in the following diagram.

Figure 1. Load balancing in one-arm mode



In the example scenario, the services Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to the virtual server Vserver-LB-1. The virtual server load balances the client request to a service. The following table lists the names and values of the entities configured on the appliance in one-arm mode.

Entity type	Name	IP address	Protocol
Virtual server	Vserver-LB-1	10.102.29.94	ANY
Services	Service-ANY-1	10.102.29.91	ANY
	Service-ANY-2	10.102.29.92	ANY
	Service-ANY-3	10.102.29.93	ANY
Monitors	TCP	None	None

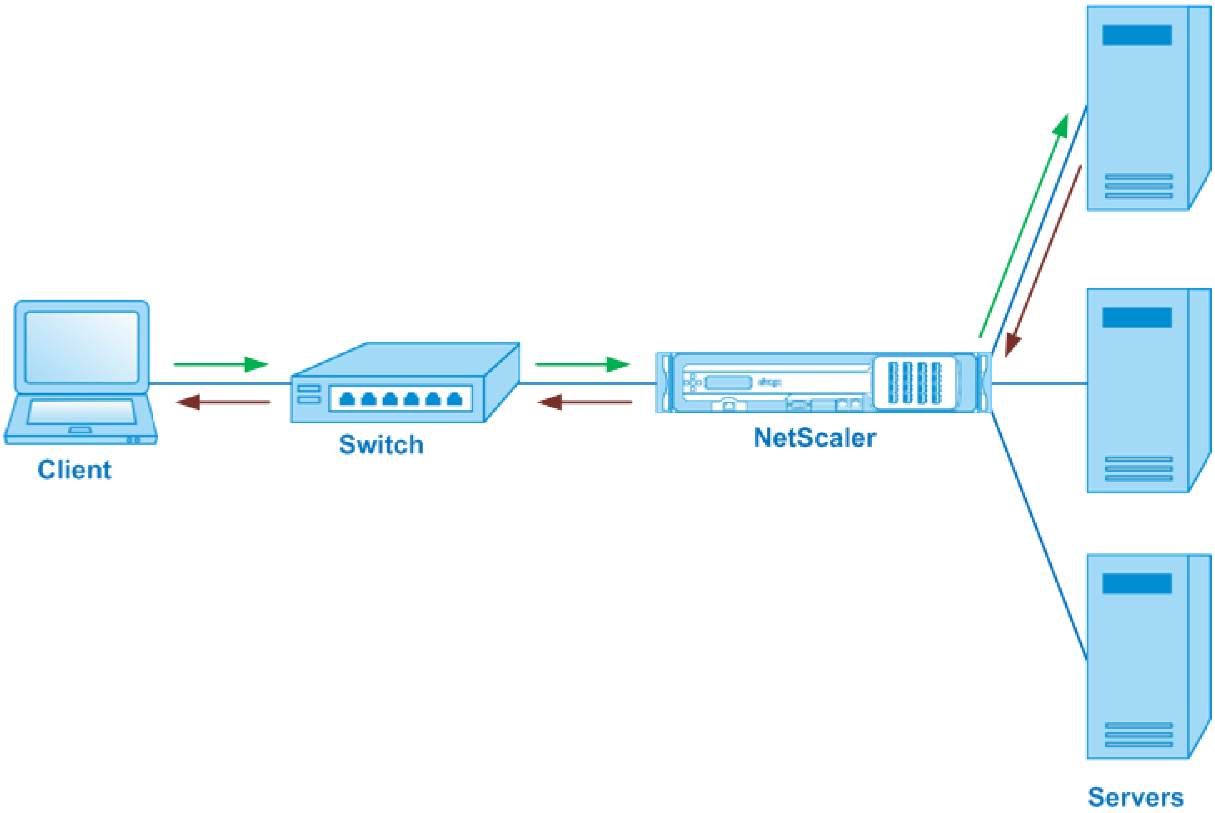
To configure a load balancing setup in one-arm mode, see "Setting Up Basic Load Balancing."

Use case 9: Configure load balancing in the inline mode

September 21, 2020

In an inline mode (also called two-arm mode) setup, you connect the NetScaler appliance to the network through multiple VLANs. The appliance receives the request from the client on one VLAN and it sends the request to the server on another VLAN. In the two-arm setup, the appliance is connected between the servers and the client. Client requests at the switch are forwarded to the appliance, and the appliance uses the configured load balancing method to select the service. This is shown in the following diagram.

Figure 1. Load balancing in inline mode



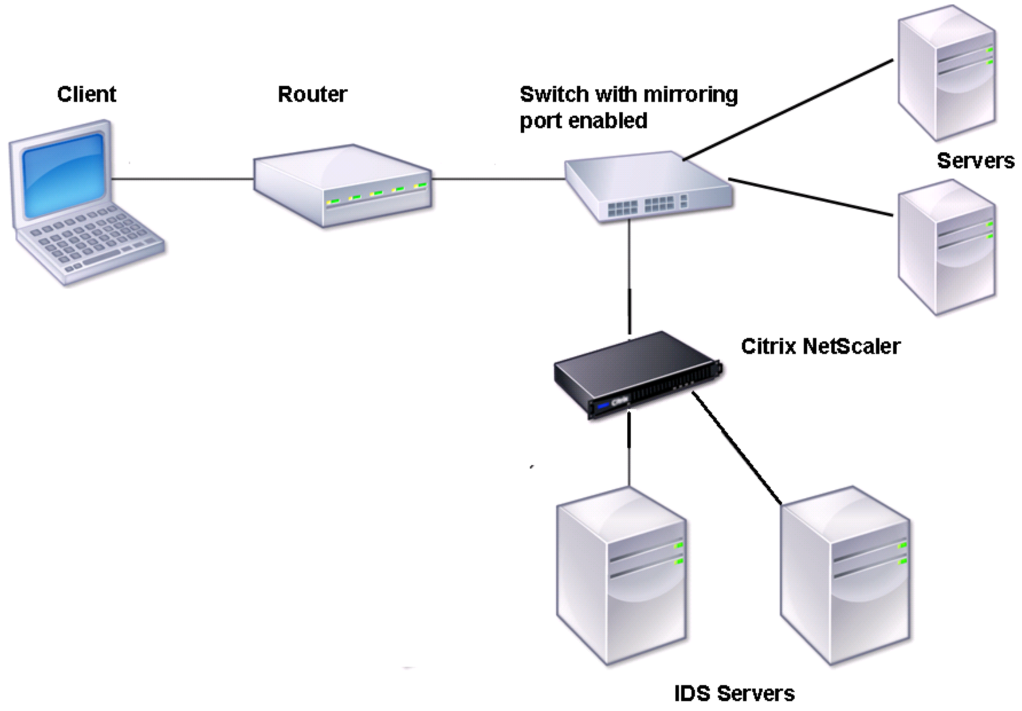
The configuration and the entity diagram for inline mode are the same as described in Configuring Load Balancing in One-arm Mode.

Use case 10: Load balancing of intrusion detection system servers

September 21, 2020

To enable the NetScaler appliance to support load balancing of intrusion detection system (IDS) servers, the IDS servers and clients must be connected through a switch that has port mirroring enabled. The client sends a request to the server. Because port mirroring is enabled on the switch, the request packets are copied or sent to the NetScaler appliance virtual server port. The appliance then uses the configured load balancing method to select an IDS server, as shown in the following diagram.

Figure 1. Topology of Load Balanced IDS Servers



Note: Currently, the appliance supports load balancing of passive IDS devices only.

As illustrated in the preceding diagram, the IDS load balancing setup functions as follows:

1. The client request is sent to the IDS server, and a switch with a mirroring port enabled forwards these packets to the IDS server. The source IP address is the IP address of the client, and the destination IP address is the IP address of the server. The source MAC address is the MAC address of the router, and the destination MAC address is the MAC address of the server.
2. The traffic that flows through the switch is mirrored to the appliance. The appliance uses the layer 3 information (source IP address and destination IP address) to forward the packet to the selected IDS server without changing the source IP address or destination IP address. It modifies the source MAC address and the destination MAC address to the MAC address of the selected IDS server.

Note: When load balancing IDS servers, you can configure the SRCIPHASH, DESTIPHASH, or SRCIPDESTIPHASH load balancing methods. The SRCIPDESTIPHASH method is recommended because packets flowing from the client to a service on the appliance must be sent to a single IDS server.

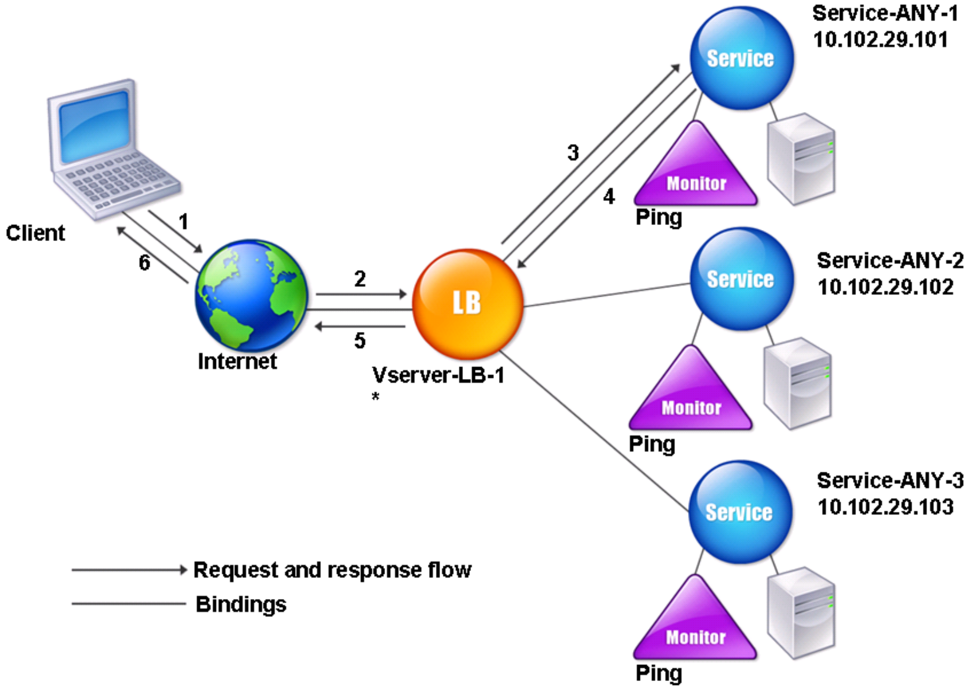
Suppose Service-ANY-1, Service-ANY-2, and Service-ANY-3 are created and bound to Vserver-LB-1. The virtual server balances the load on the services. The following table lists the names and values of the entities configured on the appliance.

Entity type	Name	IP address	Port	Protocol
Virtual server	Vserver-LB-1	*	*	ANY
Services	Service-ANY-1	10.102.29.101	*	ANY
	Service-ANY-2	10.102.29.102	*	ANY
	Service-ANY-3	10.102.29.103	*	ANY
Monitors	Ping	None	None	None

Note: You can use inline mode or one-arm mode for an IDS load balancing setup.

The following diagram shows the load balancing entities and values of the parameters to be configured on the appliance.

Figure 2. Entity Model for Load Balancing IDS Servers


Client
1
6
Internet
2
5
LB
Vserver-LB-1
*
3
4
Service
Monitor
Ping
Service-ANY-1
10.102.29.101
Service-ANY-2
10.102.29.102
Service-ANY-3
10.102.29.103
Request and response flow
Bindings

To configure an IDS load balancing setup, you must first enable MAC-based forwarding. You must also disable layer 2 and layer 3 modes on the appliance.

To enable MAC-based forwarding by using the command line interface

At the command prompt, type:

```
enable ns mode <ConfigureMode>
<!--NeedCopy-->
```

Example:

```
enable ns mode MAC
<!--NeedCopy-->
```

To enable MAC-based forwarding by using the configuration utility

Navigate to System > Settings > Configure Modes, and select MAC Based Forwarding.

Next, see “Setting Up Basic Load Balancing”, to configure a basic load balancing setup.

After you configure the basic load balancing setup, you must customize it for IDS by configuring a supported load balancing method (such as the SRCIPDESTIP Hash method on a sessionless virtual server) and enabling MAC mode. The appliance does not maintain the state of the connection and only forwards the packets to the IDS servers without processing them. The destination IP address and port remains unchanged because the virtual server is in the MAC mode.

To configure LB method and redirection mode for a sessionless virtual server by using the command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -lbMethod <LBMethodOption> -m <
    RedirectionMode> -sessionless <Value>
<!--NeedCopy-->
```

Example:

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -
    sessionless enabled
<!--NeedCopy-->
```

> **Note**
>
> For a service that is bound to a virtual server on which -m MAC option is enabled, you must bind a non-user monitor.

To configure LB method and redirection mode for a sessionless virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server and, in Redirection Mode, select MAC Based.
3. In Advanced Settings, click Methods, and select SRCIPDESTIPHASH. Click Traffic Settings, and select Sessionless Load Balancing.

To set a service to use source IP address by using the command line interface

At the command prompt, type:

```
set service <ServiceName> -usip <Value>
<!--NeedCopy-->
```

Example:

```
set service Service-ANY-1 -usip yes
<!--NeedCopy-->
```

To set a service to use source IP address by using the configuration utility

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Open a service, and in Settings select **Use Source IP Address**.

For USIP to function correctly, you must set it globally. For more information about configuring USIP globally, see "IP Addressing."

Use case 11: Isolating network traffic using listen policies

September 21, 2020

> **Note**
>
> The traffic isolation solution using shadow virtual servers to simulate multitenant isolation is no more recommended. Alternatively, Citrix recommends you to use the Citrix ADC Admin Partitioning feature for such deployments. For more information, see Admin Partitioning.

A common security requirement in a data center is to maintain network path isolation between the traffic of various applications or tenants. One application or tenant's traffic must be isolated from the traffic of other applications or tenants. For example, a financial services company would want to keep the traffic of its insurance department's applications separate from that of its financial services applications. In the past, this was easily achieved through physical separation of network service devices such as firewalls, load balancers, and IdP, and network monitoring and logical separation in the switching fabric.

As data center architectures evolve toward multitenant virtualized data centers, networking services in the aggregation layer of a data center are getting consolidated. This development has made network path isolation a critical component for network service devices and is driving the requirement for ADCs to be able to isolate traffic at the L4 to L7 levels. Furthermore, all the traffic of a particular tenant must go through a firewall before reaching the service layer.

To address the requirement of isolating the network paths, a NetScaler appliance identifies network domains and controls the traffic across the domains. The NetScaler solution has two main components: listen policies and shadow virtual servers.

Each network path to be isolated is assigned a virtual server on which a listen policy is defined so that the virtual server listens to traffic only from a specified network domain.

To isolate the traffic, listen policies can be based on several client parameters or their combinations, and the policies can be assigned priorities. The following table lists the parameters that can be used in listen policies for identifying the traffic.

Category	Parameters
Ethernet protocol	Source MAC address, destination MAC address
Network interface	Network ID, receiving throughput, sending throughput, transmission throughput
IP protocol	Source IP address, destination IP address
IPv6 protocol	Source IPv6 address, destination IPv6 address
TCP protocol	Source port, destination port, maximum segment size, payload, and other options
UDP protocol	Source port, destination port
VLAN	ID

Table 1. Client Parameters Used to Define Listen Policies

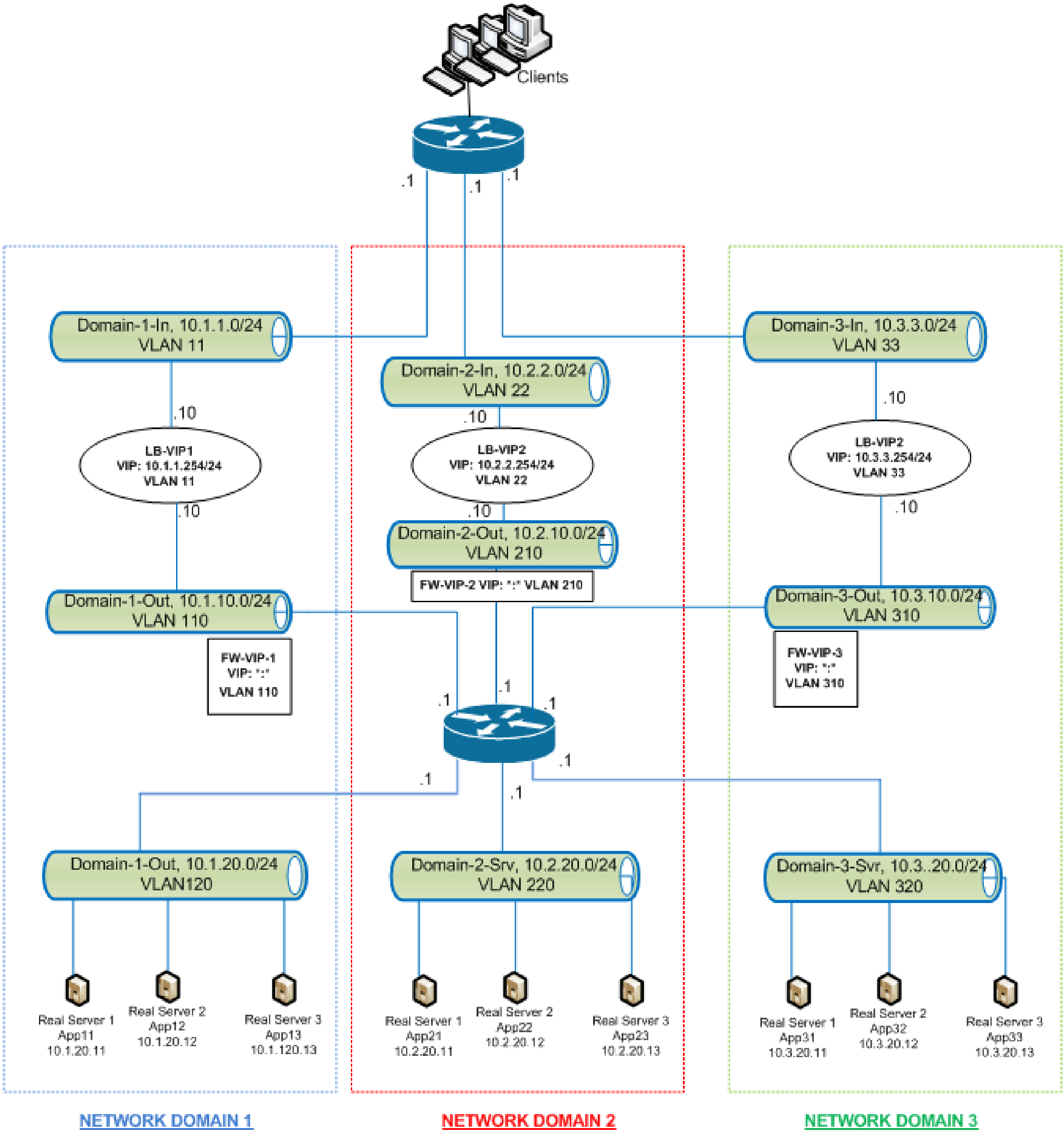
On the NetScaler appliance, a virtual server is configured for each domain, with a listen policy specifying that the virtual server is to listen only to traffic for that domain. Also configured for each domain is a shadow load balancing virtual server, which listens to traffic destined for any domain. Each of the shadow load balancing virtual servers has a wildcard (*) IP address and port, and its service type is set to ANY.

In each domain, a firewall for the domain is bound as a service to the shadow load balancing virtual server, which forwards all traffic through the firewall. Local traffic is forwarded to its destination, and traffic destined for another domain is forwarded to the firewall for that domain. The shadow load balancing virtual servers are configured for MAC mode redirection.

How network paths are isolated

The following figure shows a typical traffic flow across domains. Consider the traffic flow within Network Domain 1, and between Network Domain 1 and Network Domain 2.

Figure 1. Isolating Network Path


Clients
.1
.1
.1
Domain-1-In, 10.1.1.0/24
VLAN 11
.10
LB-VIP1
VIP: 10.1.1.254/24
VLAN 11
.10
Domain-1-Out, 10.1.10.0/24
VLAN 110
FW-VIP-1
VIP: *:*
VLAN 110
Domain-2-In, 10.2.2.0/24
VLAN 22
.10
LB-VIP2
VIP: 10.2.2.254/24
VLAN 22
.10
Domain-2-Out, 10.2.10.0/24
VLAN 210
FW-VIP-2 VIP: *:* VLAN 210
Domain-3-In, 10.3.3.0/24
VLAN 33
.10
LB-VIP2
VIP: 10.3.3.254/24
VLAN 33
.10
Domain-3-Out, 10.3.10.0/24
VLAN 310
FW-VIP-3
VIP: *:*
VLAN 310
.1
.1
.1
.1
.1
.1
Domain-1-Out, 10.1.20.0/24
VLAN120
Domain-2-Srv, 10.2.20.0/24
VLAN 220
Domain-3-Svr, 10.3..20.0/24
VLAN 320
Real Server 1
App11
10.1.20.11
Real Server 2
App12
10.1.20.12
Real Server 3
App13
10.1.120.13
Real Server 1
App21
10.2.20.11
Real Server 2
App22
10.2.20.12
Real Server 3
App23
10.2.20.13
Real Server 1
App31
10.3.20.11
Real Server 2
App32
10.3.20.12
Real Server 3
App33
10.3.20.13
NETWORK DOMAIN 1
NETWORK DOMAIN 2
NETWORK DOMAIN 3

Traffic within network domain 1

Network Domain 1 has three VLANs: VLAN 11, VLAN110, and VLAN120. The following steps describe the traffic flow.

- A client from VLAN 11 sends a request for a service available from the service pool in VLAN 120.
- The load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110. The virtual server in VLAN 110 forwards the request to shadow load balancing virtual server FW-VIP-1.

- FW-VIP-1, which is configured to listen to traffic from VLAN 110, receives the request and forwards it to VLAN 120.
- The load balancing virtual server in VLAN 120 load balances the request to one of the physical servers, App11, App12, or App13.
- The response sent by the physical server returns by the same path to the client in VLAN 11.

This configuration ensures that traffic is always segregated inside the NetScaler for all the traffic that originates from a client.

Traffic between network domain 1 and network domain 2

Network Domain 1 has three VLANs: VLAN 11, VLAN 110, and VLAN 120. Network Domain 2 also has three VLANS: VLAN 22, VLAN 210, and VLAN 220. The following steps describe the traffic flow from VALN 11 to VLAN 22.

- A client from VLAN 11, which belongs to Network Domain 1, sends a request for a service available from the service pool in VLAN 220, which belongs to the Network Domain 2.
- In Network Domain 1, the load balancing virtual server LB-VIP1, which is configured to listen to traffic from VLAN 11, receives the request and forwards the request to VLAN 110.
- Shadow load balancing virtual server FW-VIP-1, which is configured to listen to VLAN 110 traffic destined to any other domain, receives the request and forwards it to firewall virtual server FW-VIP-2 because the request is destined to a physical server in Network Domain 2.
- In Network Domain 2, FW-VIP-2 forwards the request to VLAN 220.
- The load balancing virtual server in VLAN 220 load balances the request to one of the physical servers, App21, App22, or App23.
- The response sent by the physical server returns by the same path through the firewall in Network Domain 2 and then to Network Domain 1 to reach the client in VLAN 11.

Configuration Steps

To configure network path isolation by using listen policies, do the following:

- Add listen policy expressions. Each expression specifies a domain to which traffic is destined. You can use the VLAN ID or other parameters to identify the traffic.
- For each network domain, configure two virtual servers as follows:
  - Create a load balancing virtual server for which you specify a listen policy that identifies the traffic destined for this domain. You can specify the name of an expression created earlier, or you can create a new expression while creating the virtual server.
  - Create another load balancing virtual server, referred to as shadow virtual server, for which you specify a listen policy expression that applies to traffic destined for any domain. On this virtual server, set the service type to ANY and the IP address and port to an asterisk (*). Enable MAC-based forwarding on this virtual server.

  - Enable the L2 Connection option on both the virtual servers.
    Generally, to identify a connection, the NetScaler appliance uses the 4-tuple of client IP address, client port, destination IP address, and destination port. When you enable the L2 Connection option, the Layer 2 parameters of the connection (channel number, MAC address, and VLAN ID) are used in addition to the normal 4-tuple.
- Add services representing the server pools in the domain, and bind them to the virtual server.
- Configure the firewall for each domain as a service, and bind all the firewall services to the shadow virtual server.

To isolate network traffic by using the command line interface

At the command prompt, type the following commands:

```
add policy expression <expressionName> <listenPolicyExpression>

add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -
    listenPolicy <expressionName>
<!--NeedCopy-->
```

Add a load balancing virtual server for each domain. This virtual server is for traffic of the same domain.

```
add lb vserver <name> ANY * * -l2conn ON -m MAC -listenPolicy <
    expressionName>
<!--NeedCopy-->
```

Add a shadow load balancing virtual server for each domain. This virtual server is for traffic of other domains.

Example:

```
add policy expression e110 client.vlan.id==110
add policy expression e210 client.vlan.id==210
add policy expression e310 client.vlan.id==310
add policy expression e11 client.vlan.id==11
add policy expression e22 client.vlan.id==22
add policy expression e33 client.vlan.id==33

add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -persistenceType NONE -
    listenPolicy e11
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP2 HTTP 10.2.2.254 80 -persistenceType NONE -
    listenPolicy e22
```

```
-cltTimeout 180 -l2Conn ON

add lb vserver LB-VIP3 HTTP 10.3.3.254 80 -persistenceType NONE -
    listenPolicy e33
-cltTimeout 180 -l2Conn ON


add lb vserver FW-VIP-1 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e110 -Listenpriority 1 -m MAC -cltTimeout
    120

add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e210 -Listenpriority 2 -m MAC -cltTimeout
    120

add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod
    ROUNDROBIN - listenPolicy e310 -Listenpriority 3 -m MAC -cltTimeout
    120


add service RD-1 10.1.1.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO

add service RD-2 10.2.2.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO

add service RD-3 10.3.3.1 ANY * -gslb NONE -maxClient 0 -maxReq 0 -cip
    DISABLED
-usip NO -useproxyport NO -sp ON -cltTimeout 120 -svrTimeout 120 -CKA
    NO -TCPB NO -CMP NO


bind lb vserver FW-VIP-1 RD-1

bind lb vserver FW-VIP-2 RD-2

bind lb vserver FW-VIP-3 RD-3
<!--NeedCopy-->
```

To isolate network traffic by using the configuration utility

1. Add services representing the servers, as described in Creating a Service.
2. Add each firewall as a service:
    a) Navigate to **Traffic Management > Load Balancing > Services**.
    b) Create a service, specifying the protocol as ANY, server as firewall's IP address, and port as 80.
3. Configure a load balancing virtual server.
4. Configure the shadow load balancing virtual server.
5. For each network domain, repeat steps 3 and 4.
6. From the Load Balancing Virtual Servers pane, open the virtual servers that you created and verify the settings.

Use case 12: Configure XenDesktop for load balancing

September 21, 2020

For an improved performance in the delivery of virtual desktop applications, you can integrate the NetScaler appliance with Citrix XenDesktop and use the NetScaler load balancing feature to distribute the load across the Web Interface servers and the Desktop Delivery Controller (DDC) servers.

Generally, you use XenDesktop in situations where applications are not compatible with running on a terminal server or XenApp, or if each virtual desktop has unique requirements. In such cases, you need one desktop host for each user that connects. However, the hosts can be pooled so that you need only one host for each currently connected user.

The core application service deployed for XenDesktop is the Desktop Delivery Controller (DDC). The DDC is installed on a server, and its main function is to register desktop hosts and broker client connections to them.

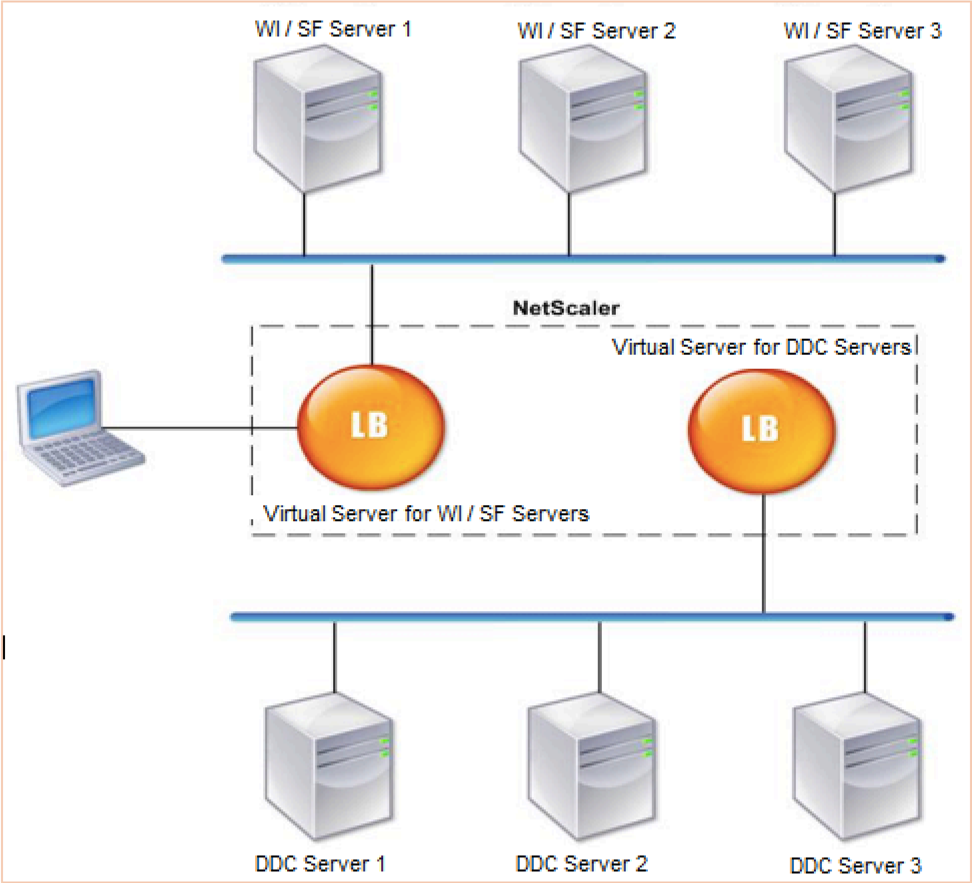
The DDC also authenticates users and manages the assembly of the users' virtual desktop environments by controlling the state of the desktops, and starting and stopping the desktops.

Generally, multiple DDCs are installed to enhance availability.

The Web Interface servers provide secure access to virtual desktops. The Web Interface is the initial connection portal to the Desktop Delivery Controller (DDC). The Web browser on the user's device sends information to the Web server, which communicates with the server farm to provide the user with access to the virtual desktop.

The following figure shows the topology of a NetScaler appliance working with XenDesktop.

Figure 1. Load Balancing of XenDesktop


WI / SF Server 1
WI / SF Server 2
WI / SF Server 3
NetScaler
Virtual Server for DDC Servers
LB
LB
Virtual Server for WI / SF Servers
DDC Server 1
DDC Server 2
DDC Server 3

> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler appliance. You can use the HTTP protocol for communication between the NetScaler and the DDC servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenDesktop by using the GUI

1. Create a service.
    a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
    b) Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
2. Create a load balancing virtual server.
    a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
    b) Create a virtual server by specifying a name, an IP address, a port, and a protocol type and

then click **OK**.

3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
    a) Click **Edit**.
    b) In the **Services and Service Groups**, click **>** and click **Add Binding**.
    c) Select the service you want to bind and enter the weight value.
    d) Click **Bind**.

To configure load balancing for XenDesktop by using the command line interface

- To create a service, at the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- To create a virtual server, at the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

Example:

add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80

- To bind a service to a load balancing virtual server, at the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

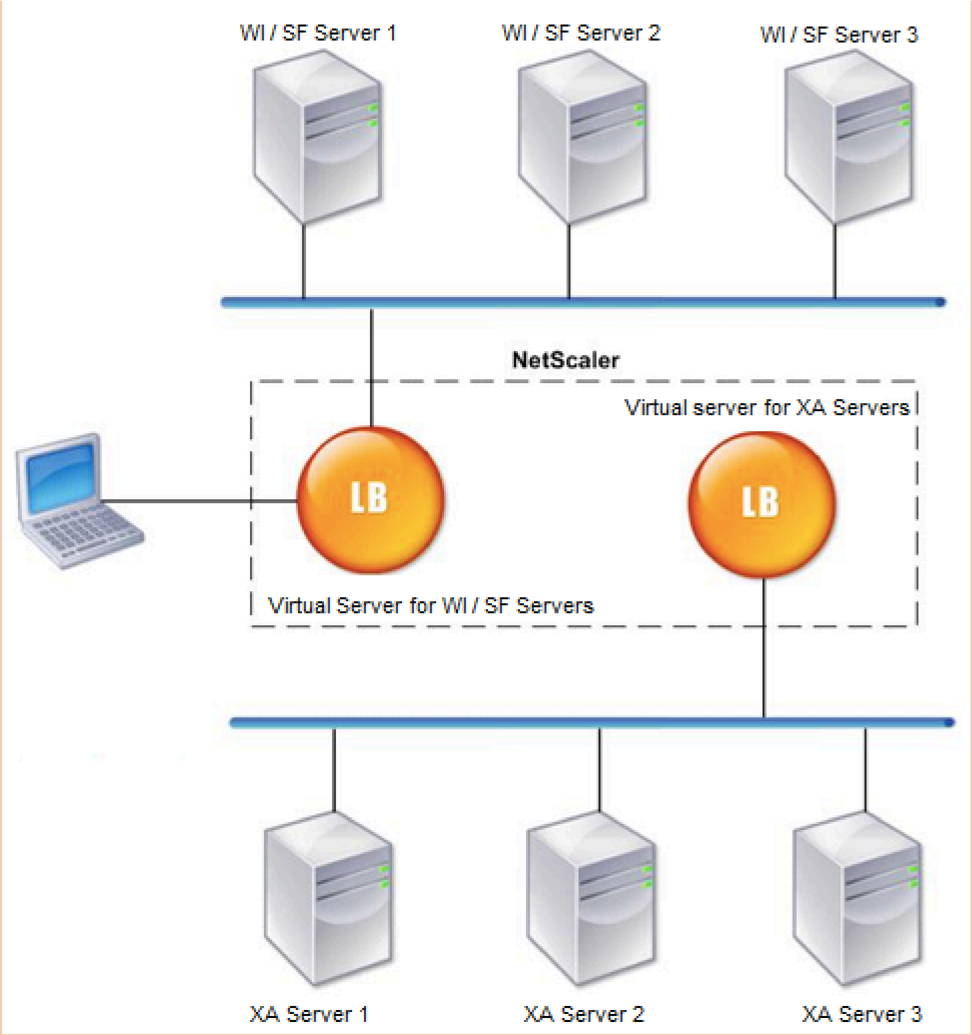
```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

Use case 13: Configure XenApp for load balancing

September 21, 2020

For efficient delivery of applications, you can integrate the NetScaler appliance with Citrix XenApp and use the NetScaler load balancing feature to distribute the load across the XenApp server farms. The following figure is a topology diagram of such a setup.

Figure 1. Load Balancing of XenApp


WI / SF Server 1
WI / SF Server 2
WI / SF Server 3
NetScaler
Virtual server for XA Servers
LB
LB
Virtual Server for WI / SF Servers
XA Server 1
XA Server 2
XA Server 3

The Web Interface servers provide secure access to XenApp application resources through the user's Web browser. The Web Interface client presents to the users all the resources, such as applications, content, and desktops that are made available in the XenApp server farms. Users can access the published resources through a standard Web browser or through the Citrix online plug-in.

The Web browser on the user's device sends information to the Web server, which communicates with the servers on the server farm to provide the user with access to the resources.

The Web Interface and the XML Broker are complementary services. The Web Interface provides users with access to applications, and the XML Broker evaluates the user's permissions to determine which applications appear in the Web Interface.

The XML service is installed on all the servers in the server farm. The XML service specified in the Web Interface functions as an XML broker. On the basis of the user credentials passed by the Web Interface server, the XML Broker server sends a list of applications accessible to the user.

In large enterprises where multiple Web Interface servers and XML Broker servers are deployed, Citrix recommends load balancing these servers by using the NetScaler appliance. Configure one virtual server to load balance all of the Web Interface servers and another for all of the XML Broker servers. The load balancing method and other features can be configured on the virtual server as required.

> **Note**
>
> Although you can use the HTTP protocol, Citrix recommends that you use SSL for communication between the client and the NetScaler. You can use the HTTP protocol for communication between the NetScaler and the WI servers even though you use the SSL protocol for communication with the client.

To configure load balancing for XenApp by using the GUI

1. Create a service.
    a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Services** and click **Add**.
    b) Create a service by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
2. Create a load balancing virtual server.
    a) Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and click **Add**.
    b) Create a virtual server by specifying a name, an IP address, a port, and a protocol type and then click **OK**.
3. Bind the service to the load balancing virtual server.
4. Navigate to **Configuration** > **Traffic Management** > **Load Balancing** > **Virtual Servers** and select a server.
    a) Click **Edit**.
    b) In the **Services and Service Groups**, click **>** and click **Add Binding**.
    c) Select the service you want to bind and enter the weight value.
    d) Click **Bind**.

To configure load balancing for XenApp by using the command line interface

- To create a service, at the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service Service-HTTP-1 192.0.2.5 HTTP 80
<!--NeedCopy-->
```

- To create a virtual server, at the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
<!--NeedCopy-->
```

- To bind a service to a load balancing virtual server, at the command prompt, type:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
<!--NeedCopy-->
```

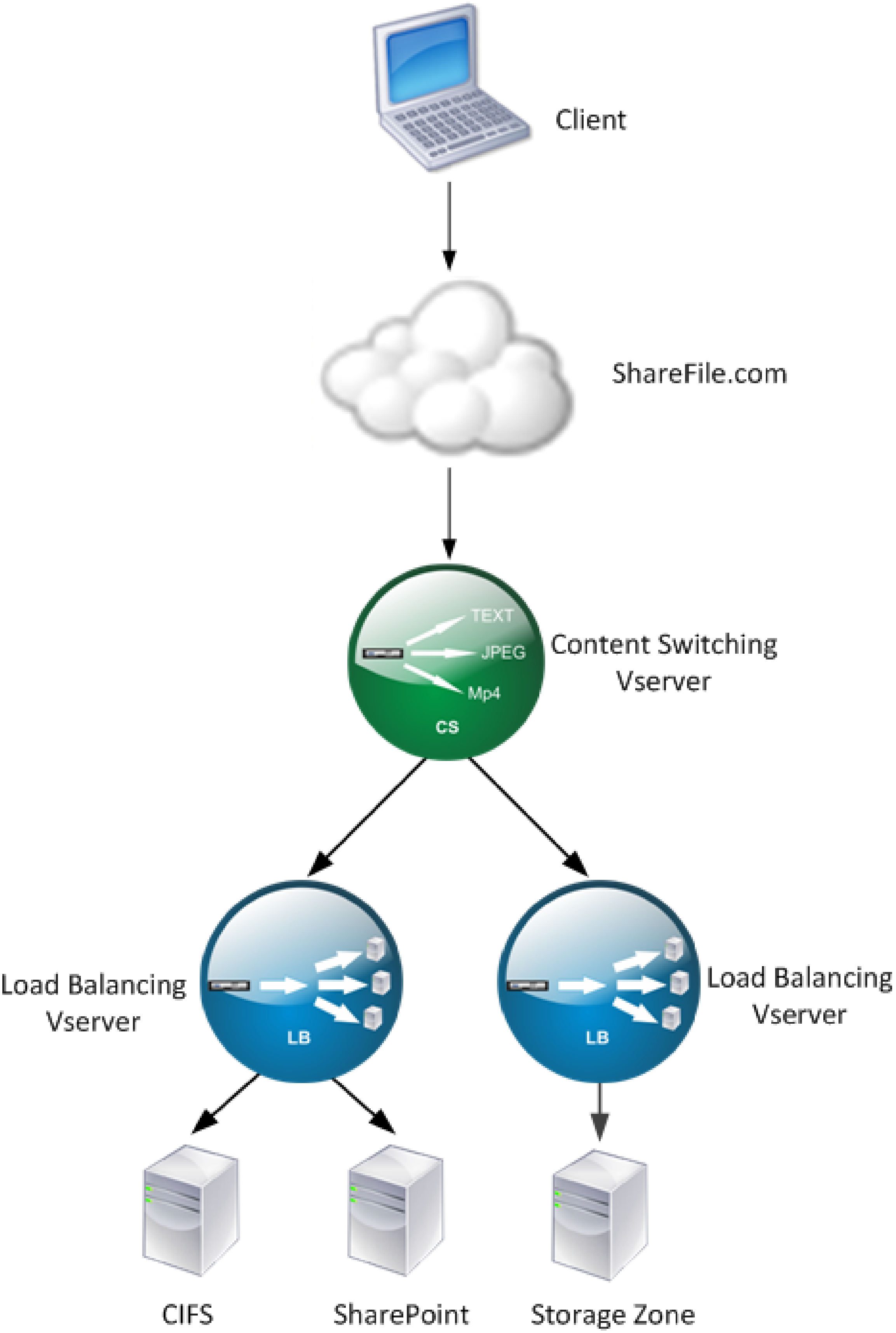
Use case 14: ShareFile wizard for load balancing Citrix ShareFile

September 21, 2020

You can configure load balancing for Citrix ShareFile using the wizard. The Citrix ShareFile wizard helps in setting up load balancing configuration for ShareFile site based on the type of content requested. The content switching server directs the request based on whether it is a StorageZone, CIFS or a SharePoint request. The content switching is based on policies. The wizard auto generates the policies to identify whether the request is for StorageZone, CIFS or SharePoint. The content switching virtual server uses these policies to direct the request to the correct load balancing server.

A typical data flow can be depicted as shown in the diagram below.

Figure 1. ShareFile Data Load Balancing

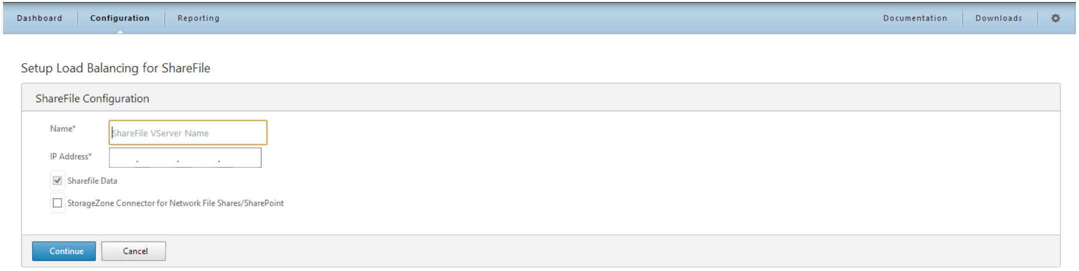

Client
ShareFile.com
TEXT
JPEG
Mp4
CS
Content Switching Vserver
Load Balancing Vserver
LB
LB
Load Balancing Vserver
CIFS
SharePoint
Storage Zone

You can view the load balancing virtual servers that the ShareFile wizard creates by navigating to Traffic Management >Virtual Servers and Services > Virtual Servers. You cannot manually remove the virtual servers created using the ShareFile wizard. Use the wizard to remove the virtual servers.

NetScaler uses the LDAP authentication for SharePoint or CIFS request. Hash authentication is used for authenticating requests for StorageZones.

To configure a NetScaler appliance for load balancing Citrix ShareFile

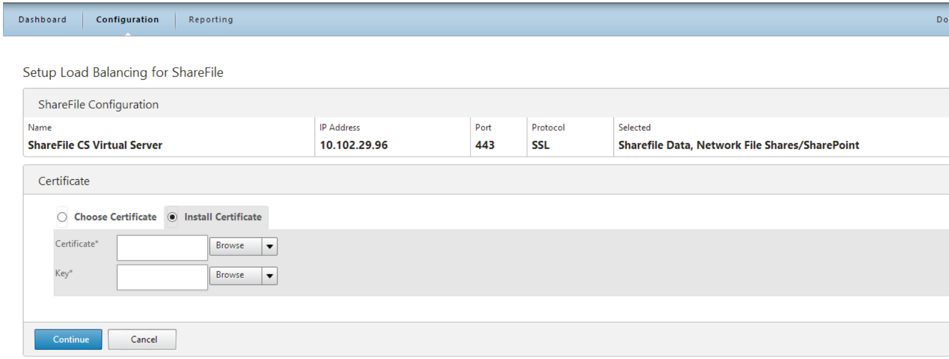
1. In the navigation pane, click Load Balancing.
2. Navigate to Traffic Management > Load Balancing.
3. Under Citrix ShareFile, click Setup NetScaler for ShareFile.
4. On the Setup Load Balancing for ShareFile page, provide the following information:
    - Name: Name of the content switching virtual server.
    - IP Address: IP address of the content switching virtual server.
    - If you want to setup load balancing for CIFS or SharePoint, click the StorageZone Connector for Network File Shares/SharePoint checkbox and then click Continue. By default ShareFile Data checkbox is selected.


Dashboard
Configuration
Reporting
Documentation
Downloads
Setup Load Balancing for ShareFile
ShareFile Configuration
Name*
ShareFile VServer Name
IP Address*
Continue
Cancel

☑ Sharefile Data

☐ StorageZone Connector for Network File Shares/SharePoint

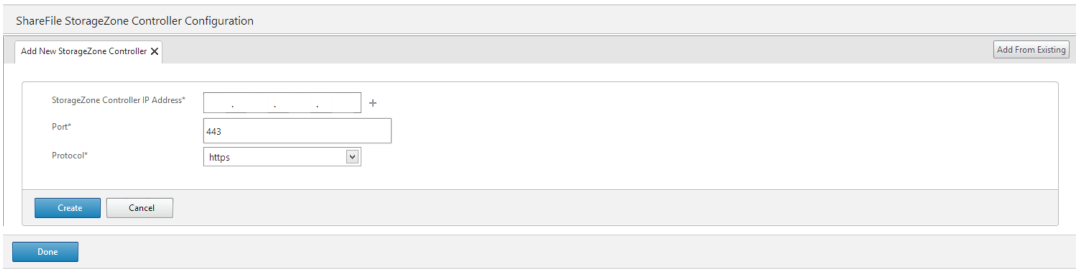
5. Provide a valid certificate. If you have a certificate, click Choose Certificate and from the drop-down list select the certificate. If you have to install a certificate, click Install Certificate and provide the Certificate-Key pair.


Dashboard
Configuration
Reporting
Do
Setup Load Balancing for ShareFile
ShareFile Configuration
Certificate
Certificate*
Browse
Key*
Browse
Continue
Cancel

Name	IP Address	Port	Protocol	Selected
ShareFile CS Virtual Server	10.102.29.96	443	SSL	Sharefile Data, Network File Shares/SharePoint

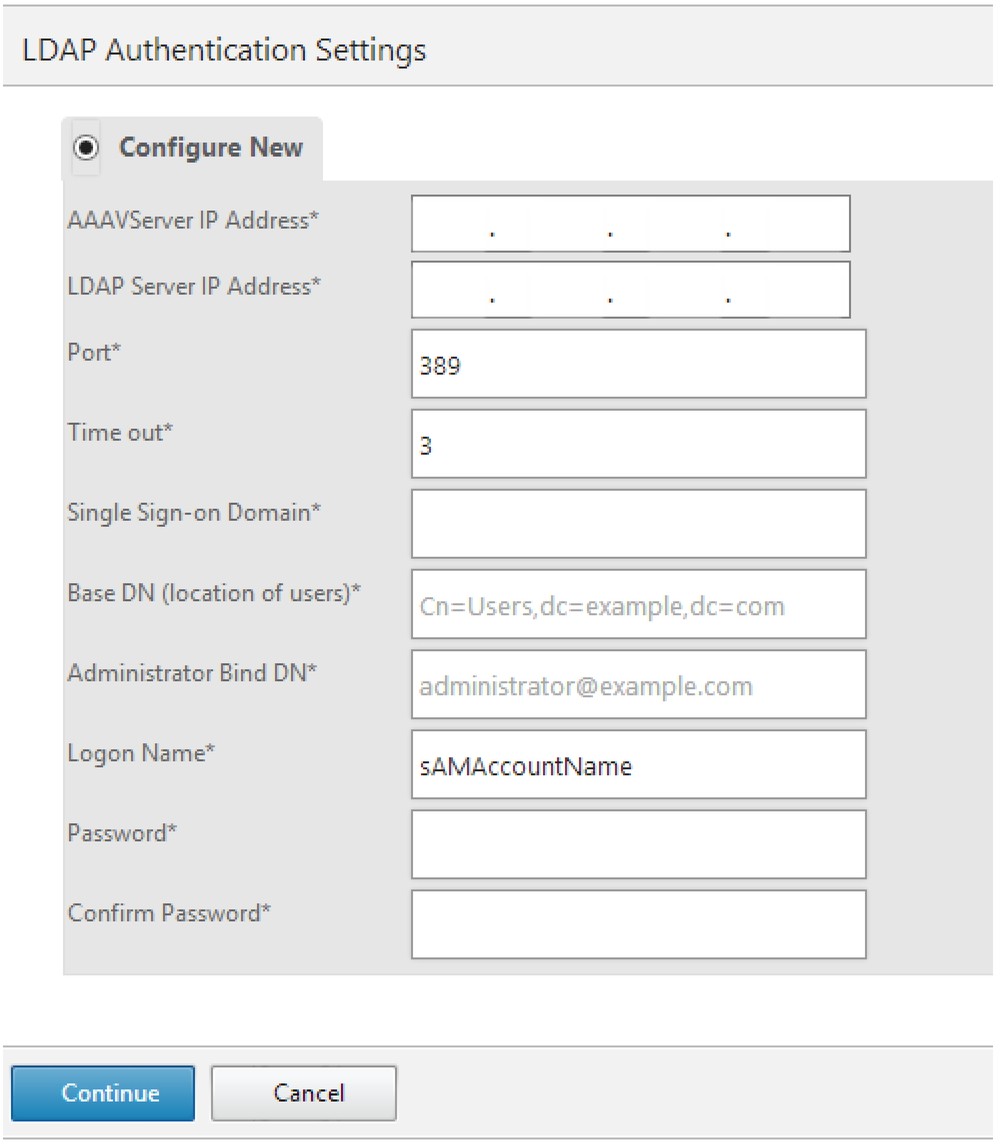
○ Choose Certificate ● Install Certificate

6. Click Continue.
7. In the Add New StorageZone Controller dialog box, specify the values of the following parameters:
    - StorageZone Controller IP Address— IP address
    - Port— Port number. The default value is 443.
    - Protocol— Select from HTTPS or HTTP


ShareFile StorageZone Controller Configuration
Add New StorageZone Controller ✕
Add From Existing
Create
Cancel
Done

StorageZone Controller IP Address*
Port* 443
Protocol* https

8. Click Create and then click Done. The wizard automatically creates a service and autogenerate the name of the service.
9. If you chose load balancing for CIFS or SharePoint in step 4.c, then specify the values for LDAP Authentication Settings:
   - AAAVServer IP Address— IP address of AAA virtual server
   - LDAP Server IP Address— IP Address of the LDAP server
   - Port— Port number. The default value is 389
   - Time out— The time out value in minutes
   - Single Sign-on Domain— Single sign-on domain name
   - Base DN— Base domain name
   - Administrator Bind DN— LDAP account name with the domain name, for example, administrator@domainname.com
   - Logon Name— Logon name is the sAMAccount name
   - Password and Confirm Password— Enter the password and confirm the password



LDAP Authentication Settings

● Configure New

AAAVServer IP Address*	. . .
LDAP Server IP Address*	. . .
Port*	389
Time out*	3
Single Sign-on Domain*	
Base DN (location of users)*	Cn=Users,dc=example,dc=com
Administrator Bind DN*	administrator@example.com
Logon Name*	sAMAccountName
Password*	
Confirm Password*	

10. Click Continue and then click Done.

To remove load balancing configuration for ShareFile

1. Click on Configuration > Traffic Management > Load Balancing.
2. On the Load Balancing page, under Citrix ShareFile click on Remove ShareFile Configuration.

Troubleshooting

September 21, 2020

If the load balancing does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for Troubleshooting Load Balancing

For best results, use the following resources to troubleshoot a content switching issue on a NetScaler appliance:

- Latest ns.conf file
- Relevant newnslog files
- Ethereal packet traces recorded on the appliance and relevant client, if possible
- The ns.log file

In addition to the above resources, the following tools expedite troubleshooting:

- A browser add-on tool that can display HTTP headers. This can be used to troubleshoot persistency related issues.
- The Wireshark application customized for the NetScaler trace files.

Troubleshooting Load Balancing Issues

- **Issue**

CPU usage reaches 100% when a user monitor is bound to a service that is bound to a virtual server on which -m MAC option is enabled.

- **Resolution**

Bind a non-user monitor to the service.

- **Issue**

  I created a user script for monitoring, but it is not working.

  Resolution

  Check the number of arguments in the script. The limit is 512. A script with more than 512 arguments might not work properly. Use the nsumon-debug.pl script from the CLI to debug the script.

- **Issue**

  I see a lot of monitor probes, and they seem to be increasing the network traffic unnecessarily. Is there a way to turn off the monitor probes?

  Resolution

  You can turn off the monitor probe connections, by disabling the monitor or setting the value of the healthMonitor parameter in the set service command to NO. With the NO option, the appliance shows the service as UP at all times.

- **Issue**

  I have set up monitors for services, but connections are still directed to servers that are DOWN.

  Resolution

  You probably need to decrease the monitor probe intervals. The NetScaler appliance does not detect the DOWN state until the monitor sends a probe.

- **Issue**

  A metric bound to the monitor is present in the local and custom metric tables.

  Resolution

  Add the local prefix to the metric name if the metric is chosen from the local metric table. However, if the metric is chosen from the custom table, you don't need to add any prefix.

- **Issue**

  The monitor probes to a service are not reaching the service.

  Resolution

  Check whether you have set a limit on the number of connections for a service. If yes, exempt monitor-probe connections from this limit by setting the monitorSkipMaxClient parameter to ENABLED.

- **Issue**

  I am able to ping the servers, but the state of the services is always shown as DOWN.

  Resolution

  Check the type of monitors configured. For example, if a server is not configured for SSL and you use an HTTPS monitor, the state of the service is marked as DOWN. In this case using a TCP monitor should change the state of the service to UP.

- **Issue**

  Setting a weight for load monitors does not help in deciding the state of the service.

  Resolution

  Load monitors cannot decide the state of the service. Therefore, setting a weight on the load monitors is inappropriate.

- **Issue**

  A service is not stable.

  Resolution

  Consider troubleshooting the following components:

  - Verify that a correct server is bound to the service.
  - Verify the type of monitor bound to the service.
  - Verify the reasons for the monitor failures. You can open service from the Services page and verify the details for the number of probes, failures, and last response status for the monitor in the Monitors tab of the Configure service dialog box. To display the details, click the monitor configured.
  - If it is a custom monitor, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.
  - You can record packet traces on the NetScaler appliance and verify the monitor probes and server response for further investigation.

- **Issue**

  The virtual IP (VIP) address is not stable or its status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:

  - Verify that the load balancing feature is licensed.
  - Verify that the feature is enabled.
  - Verify that an appropriate service is bound to the virtual server.
  - If the status of the VIP address is displayed as DOWN, verify that an administrator has enabled the service. If it is not, the status of the service should be Out-Of-Service. In such as case, you must enable the service and verify if the issue is resolved.
  - Verify the service(s) bound to the virtual server and complete the troubleshooting steps mentioned for service not stable issue.
  - If the VIP address is not stable, all the services bound to the virtual server should fail. Therefore, verify if all the services are failing at the same time. If it is so, there is a network issue between the NetScaler appliance and the servers.

- **Issue**

  The site is experiencing uneven load balancing.

  Resolution

  Consider troubleshooting the following components:

  - Verify the load balancing method configured on the appliance.

  - Verify weights associated with the services are as expected.

  - If the load balancing method is other than round robin, verify the number of connections to the server logged in the newnslog file. You can run the following command to verify the number on the newnslog file:

```
    ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

    Verify the services for the specific virtual server and check for the Response time, Open Established connections (OE), Hits, Persistent Hits and persistent rate (P) to troubleshoot the issue further.

  - If the load balancing method is round robin, verify the persistent Hits as mentioned in the preceding step. Additionally, verify if the service is not stable. If it is not, complete the troubleshooting steps mentioned for service not stable issue
  - Verify if persistency is configured on the appliance.
  - Verify is any service is not stable. If yes, complete the troubleshooting steps mentioned for service not stable issue.

- **Issue**

  The service status is displayed as DOWN.

  Resolution

  Consider troubleshooting the following components:

  - Verify whether a SNIP address is configured.
  - Verify that appropriate monitors are bound to the service.
  - If custom monitors are bound to the service, bind a TCP or ping monitor to the service and verify the status of the monitor. If this resolves the issue, there is some problem with the custom monitor and the monitor requires further investigation.
  - Verify if the status of service is displayed as DOWN for the server that is in another subnet. If yes, verify if Use Subnet IP (USNIP) resolves the issue because this could be due to the MIP address being unable to communicate to the server.

- **Issue**

  There is an issue with the response time.

  Resolution

  Consider troubleshooting the following components:

  - Verify the server response time from the service stats either by running the following command:

    ```
    ## nsconmsg -K <newnslog_file> -s ConLb=2 -d oldconmsg
    ```

  - Check for service not stable and service status being displayed as DOWN issues.

- **Issue**

  One of the servers is serving more requests than the other load balanced servers.

  Resolution

Consider troubleshooting the following components:

  - Verify the load balancing method. Use the round robin method to distribute the client request equally regardless of the load on the servers.
  - Determine whether persistence is enabled for the load balancing configuration. If persistence is enabled, a given servers might be carrying a heavier load to maintain its session, especially If the persistence sessions are long.
  - Verify whether weights are assigned to each service. Assigning proper weights helps in proper load distribution.

- **Issue**

  Connections to a specific load balanced server are stalled. For example, all connections to one Outlook server might be stalled.

  Resolution

  Consider troubleshooting the following components:

  - Verify the load balance method. If it is round robin, consider changing the method to least connections.
  - Consider reducing the monitor time-out period. A shorter timeout period helps in marking a service as DOWN sooner, which would help in directing the traffic to server which is functional.
  - If the connections are stalled for a long period, surge-queue might build. Consider flushing the surge-queue to avoid a sudden spike in load on the server.
  - If the servers are working at their maximum level, consider adding a new server for better performance.

- **Issue**

  A majority of the connections are directed to a particular server, even when the least connections method for load balancing is configured.

  Resolution

  Determine whether persistence is configured and is of type source IP. If source IP persistence is configured even with the least connections method, the requests go to a specific server. The server's IP address is required for maintaining the session information. Consider using HTTP Cookies based persistence.

- **Troubleshooting Tips**
  For other issues, consider following tips to troubleshoot an issue not listed above:
  - If multiple load monitors are bound to a service, the load on the service is the sum of all the values on the load monitors bound to it. For load balancing to work properly, you must bind the same set of monitors to all the services.

- If you disable a load monitor bound to the service and the service is bound to a virtual server, the virtual server uses the round robin method for load balancing.
- When you bind a service to a virtual server where the load balancing method is CUSTOMLOAD and the service status is UP, the virtual server uses the initial round robin method for load balancing. It continues to be in round robin if the service has no custom load monitors, or if status of at least one of the custom load monitors is not UP.
- All the services that are bound to a virtual server where the load balancing method is CUSTOMLOAD, the services must have load monitors bound to them.
- The CUSTOMLOAD load balancing method also follows startup round robin.
- If you disable a metric-based binding and this is the last active metric, the specific virtual server uses the round robin method for load balancing. A metric is disabled by setting the metric threshold to zero.
- When a metric bound to a monitor crosses the threshold value, that particular service is not considered for load balancing. If all the services have reached the threshold, the virtual server uses the round robin method for load balancing and an error message "5xx - server busy error" is displayed.
- A maximum of 10 metrics from a custom table can be bound to the monitor.
- The OIDs must be scalar variables.
- For successful load balancing, the interval must be as low as possible. If the interval is high, the time period for retrieving the load value increases. As a result, load balancing takes place using improper values.
- A user cannot modify the local table.

Load balancing FAQs

September 21, 2020

What are the various load balancing policies I can create on the NetScaler appliance

You can create the following types of load balancing policies on the NetScaler appliance:

```
-  Least Connections
-  Round Robin
-  Least response time
-  Least bandwidth
-  Least packets
-  URL hashing
-  Domain name hashing
-  Source IP address hashing
-  Destination IP address hashing
-   Source IP - Destination IP hashing
-   Token
-   LRTM
```

Can I achieve the Web farm security by implementing load balancing using the NetScaler appliance?

Yes. You can achieve Web farm security by implementing load balancing using the NetScaler appliance. NetScaler appliance enables you to implement the following options of the load balancing feature:

```
-   IP Address hiding: Enables you to install the actual servers to be on private IP address space for security reasons and for IP address conservation. This process is transparent to the end-user because the NetScaler appliance accepts requests on behalf of the server. While in the address hiding mode, the appliance completely isolates the two networks. Therefore, a client can access a service running on the private subnet, such as FTP or a Telnet server, through a different VIP on the appliance for that service.
-   Port Mapping: Enables the actual TCP services to be hosted on non-standard ports for security reasons. This process is transparent to the end-user as the NetScaler appliance accepts requests on behalf of the server on the standard advertised IP address and port number.
```

What are various devices that I can use to load balance with a NetScaler appliance?

You can load balance following devices with a NetScaler appliance:

```
-   Server farms
-   Caches or Reverse Proxies
-   Firewall devices
-   Intrusion detection systems
-   SSL offload devices
-   Compression devices
-   Content Inspection servers
```

Why should I implement the load balancing feature for the website?

You can implement the Load balancing feature for the website to take the following advantages:

```
- Reduce the response time: When you implement the load balancing feature for the website, one of the major benefits is the boost you can look forward to in load time. With two or more servers sharing the load of the web traffic, each of the servers runs less traffic load than a single server alone. This means there are more resources available to fulfill the client requests. This results in a faster website.
- Redundancy: Implementing the load balancing feature introduces a bit of redundancy. For example, if the website is balanced across three servers and one of them does not respond at all, the other two can keep running and the website visitors do not even notice any downtime. Any load balancing solution immediately stops sending traffic to the backend server that is not available.
```

Why do I need to disable the Mac Based Forwarding (MBF) option for Link Load Balancing (LLB)?

```
- If you enable the MBF option, the NetScaler appliance considers that the incoming traffic from the client and the outgoing traffic to the same client flow through the same upstream router. However, the LLB feature requires that the best path be chosen for the return traffic.
- Enabling the MBF option breaks this topology design by sending the outgoing traffic through the router that forwarded the incoming client traffic.
```

Networking

September 21, 2020

The following topics provide a conceptual reference and instructions for configuring the various networking components on the NetScaler appliance.

IP Addressing	Learn the various types of NetScaler owned IP addresses and how to create, customize, and remove them.

Interfaces	Configure some of the basic network configurations that must be done to get started.
Access Control Lists (ACLs)	Configure the different types of Access Control Lists and how to create, customize, and remove them.
IP Routing	Learn and configure the routing functionality of the NetScaler appliance, both static and dynamic.
Internet Protocol version 6 (IPv6)	Learn how the NetScaler appliance supports IPv6.
Traffic Domains	Learn and configure traffic domains to segment network traffic for different applications.
VXLAN	Learn and configure VXLANs to meet the scalability needs in your datacenter.

IP Addressing

September 21, 2020

Before you can configure the NetScaler appliance, you must assign the NetScaler IP Address (NSIP), also known as the Management IP address. You can also create other NetScaler-owned IP addresses for abstracting servers and establishing connections with the servers. In this type of configuration, the appliance serves as a proxy for the abstracted servers. You can also proxy connections by using network address translations (INAT and RNAT). When proxying connections, the appliance can behave either as a bridging (Layer 2) device or as a packet forwarding (Layer 3) device. To make packet forwarding more efficient, you can configure static ARP entries. For IPv6, you can configure neighbor discovery (ND).

Configuring NetScaler-Owned IP Addresses

September 21, 2020

The NetScaler-owned IP Addresses—NetScaler IP Address (NSIP), Virtual IP Addresses (VIPs), Subnet IP Addresses (SNIPs), and Global Server Load Balancing Site IP Addresses (GSLBIPs)—exist only on the NetScaler appliance. The NSIP uniquely identifies the NetScaler on your network, and it provides access to the appliance. A VIP is a public IP address to which a client sends requests. The NetScaler terminates the client connection at the VIP and initiates a connection with a server. This new connection uses a SNIP or a MIP as the source IP address for packets forwarded to the server. If you have multiple data centers that are geographically distributed, each data center can be identified by a unique GSLBIP. You can configure some NetScaler-owned IP addresses to provide access for management applications.

Configuring the NetScaler IP Address (NSIP)

September 21, 2020

The NetScaler IP (NSIP) address is the IP address at which you access the NetScaler appliance for management purposes. The appliance can have only one NSIP, which is also called the management IP address. You must add this IP address when you configure the NetScaler for the first time. You cannot remove an NSIP address. For security reasons, the NSIP should be a non-routable IP address on your organization's LAN.

If you modify this address, you must reboot the NetScaler appliance. If the subnet address of the new NSIP address is different from the previous one, you must add a default route for this subnet so that the new NSIP address becomes reachable from other networks on the LAN.

> **Important**
>
> Configuring the NetScaler IP address is mandatory.

Changing the NSIP address of a NetScaler appliance consists of the following tasks:

- Change the NSIP address.
- Add a default route for the subnet address of the NSIP address, if one is not present.
- Save the configuration.
- Restart the appliance.

Command Line Procedures

To change the NetScaler IP address by using the NetScaler command line:

At the command prompt, type:

- **set ns config -IPAddress** <ip_addr> **-netmask** <netmask>
- **show ns config**

To add a default route by using the NetScaler command line:

At the command prompt, type:

- **add route 0 0** <gateway IP address>
- **show route**

To save the configuration by using the NetScaler command line:

At the command prompt, type:

- **save config**

To restart the NetScaler appliance by using the NetScaler command line:

At the command prompt, type:

- **reboot**

GUI Procedures

To configure the NetScaler IP address by using the NetScaler GUI:

1. Click the gear icon in the top-right corner of the **Configuration** page.
2. Click the** NetScaler IP address** pane.
3. On the **NetScaler IP Address** page, set the following parameters, and then click **Done**:
   - NetScaler IP Address
   - Netmask

To add a default route by using the NetScaler GUI:

Navigate to** System** >** Network** >** Routes** and, on the** Basic** tab, add a default route with the following parameter settings, and then click **Create**.

- Network (set to zero)
- Netmask (set to zero)
- Gateway (IP address of the gateway)

To restart the NetScaler by using the NetScaler GUI:

1. On the **System Information** tab page of the **System** node, click **Reboot**.
2. When prompted to reboot, select **Save Configuration** to make sure that you do not lose any configurations.

Sample configuration

In the following example, the NSIP address of a NetScaler appliance is changed to 192.0.2.90, which has a different subnet address (192.0.2.0/24) than the previous NSIP address. Therefore, a default route is added for this subnet, so that the new NSIP address becomes reachable from other networks.

```
> set nsconfig -ipAddress 192.0.2.90 -netmask 255.255.255.0

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> add route 0 0 192.0.2.1

 Warning: The configuration must be saved and the system rebooted for
    these settings to take effect
> save config

 Done
> reboot
```

Configuring and Managing Virtual IP (VIP) Addresses

September 21, 2020

Configuration of a virtual server IP (VIP) address is not mandatory during initial configuration of the NetScaler appliance. When you configure load balancing, you assign VIP addresses to virtual servers.

For more information about configuring a load balancing setup, see Load Balancing.

In some situations, you need to customize VIP attributes or enable or disable a VIP address. A VIP address is usually associated with a virtual server, and some of the VIP attributes are customized to meet the requirements of the virtual server. You can host the same virtual server on multiple NetScaler appliances residing on the same broadcast domain, by using ARP and ICMP attributes. After you add a VIP (or any IP address), the NetScaler sends, and then responds to, ARP requests. VIPs are the only NetScaler-owned IP addresses that can be disabled. When a VIP address is disabled, the virtual server using it goes down and does not respond to ARP, ICMP, or L4 service requests. As an alternative to creating VIP addresses one at a time, you can specify a consecutive range of VIP addresses.

To create a VIP address by using the NetScaler command line:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.59 255.255.255.0 -type VIP
 Done
<!--NeedCopy-->
```

To create a range of VIP addresses by using the NetScaler command line:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type <type>
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.[60-64] 255.255.255.0 -type VIP
ip "10.102.29.60" added
ip "10.102.29.61" added
ip "10.102.29.62" added
ip "10.102.29.63" added
ip "10.102.29.64" added
 Done
<!--NeedCopy-->
```

To enable or disable an IPv4 VIP address by using the NetScaler command line:

At the command prompt, type one of the following sets of commands to enable or disable a VIP and verify the configuration:

- enable ns ip <IPAddress>
- show ns ip <IPAddress>
- disable ns ip <IPAddress>
- show ns ip <IPAddress>

Example:

```
> enable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Enabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled
 Done
> disable ns ip 10.102.29.79
 Done
> show ns ip 10.102.29.79

        IP: 10.102.29.79
        Netmask: 255.255.255.255
        Type: VIP
        state: Disabled
        arp: Enabled
        icmp: Enabled
        vserver: Enabled
        management access: Disabled
          telnet: Disabled
          ftp: Disabled
          ssh: Disabled
          gui: Disabled
          snmp: Disabled
        Restrict access: Disabled
        dynamic routing: Disabled
        hostroute: Disabled

 Done
<!--NeedCopy-->
```

To configure a VIP address by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPs** > **IPV4s**, and add a new IP address or edit an existing address.

To create a range of VIP addresses by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. In the **Action** list, select **Add Range**.

To enable or disable a VIP address by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Do one of the following:
    - Select a VIP address.
    - Hold down the **Ctrl** key and select multiple server address entries.
    - Hold down the **Shift** key and select a range of server address entries.
    - Select all the addresses by selecting the checkbox on the left side of the header row.
3. From the **Action** list, select **Disable** or **Enable**.

Detecting a NetScaler Appliance in a UDP Load Balancing Setup through TTL Updates

The following table displays how a NetScaler appliance handles the TTL value of received packets in different functionalities.

Functionality	TTL value
Virtual Server	TTL is set to 255 when forwarding the request to the backend servers. TTL is decremented by 1 when forwarding the response to the client.
L2 Mode	TTL is not changed.
L3 Mode	TTL is set to 255.
INAT	TTL is set to 255 when forwarding the request to the backend server. TTL is decremented by 1 when forwarding the response to the client.

Some enterprises/scenarios running a monitoring application requires the NetScaler appliance of a load balancing setup to be detected as one of the hop in a traceroute. A NetScaler appliance of a load balancing setup is not detected in a traceroute because the appliance, by default, sets the TTL value to 255 instead of decrementing it when forwarding the request to a backend server.

To meet this requirement, **Decrement TTL** parameter of a VIP address can be used. This parameter applies to all UDP virtual servers using this VIP.

When you enable the **Decrement TTL **parameter of a VIP, the NetScaler appliance decrements the TTL value by 1 instead of setting it to 255 when forwarding requests, which are received on the UDP virtual servers that uses this VIP.

Monitoring applications using traceroute data can now detect the presence of a NetScaler appliance of a UDP load balancing setup.

Before You Begin

Before you begin configuring a NetScaler appliance to be detected in a traceroute of a load balancing setup, note the following points:

- Decrement TTL parameter is supported only for UDP load balancing virtual servers.
- Decrement TTL parameter is supported for IPv4 VIP as well as IPv6 VIP (VIP6) addresses.
- Decrement TTL parameter is supported for standalone NetScaler appliances as well as for high availability (HA) and cluster setups.

Configuration Steps

Configuring a NetScaler appliance to be detected in a traceroute of a UDP load balancing setup consists of the following tasks:

- Create a UDP load balancing configuration
- Enable the Decrement TTL parameter for the VIP address

NetScaler Command Line Procedures

To enable the decrement TTL option for a VIP address by using the NetScaler command line:

- To enable the decrement TTL option for a VIP address while adding the VIP address, at the command prompt, type:
  - **add ns ip** <ip> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>
- To enable the decrement TTL option for an existing VIP address, at the command prompt, type:
  - **set ns ip** <ip> <mask> -**decrementTTL ENABLED**
  - **show ns ip** <VIP address>

To enable the decrement TTL option for a VIP6 address by using the NetScaler command line:

- To enable the decrement TTL option for a VIP6 address while adding the VIP6 address, at the command prompt, type:
  - **add ns ip6** <IP6/prefix> <mask> -**type VIP** -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6/prefix>
- To enable the decrement TTL option for an existing VIP6 address, at the command prompt, type:
  - **set ns ip6** <ip6/prefix> <mask> -**decrementTTL ENABLED**
  - **show ns ip6** <VIP6 address>

```
> add ns ip 203.0.113.30 -type VIP -decrementTTL ENABLED
 Done

> add ns ip6  2001:DB8:5001::30 -type VIP -decrementTTL ENABLED
 Done
```

NetScaler GUI Procedures

To enable the decrement TTL option for a VIP address by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPs** > **IPv4s**, and enable the **Decrement TTL** parameter while adding a new VIP address or editing an existing address.

To enable the decrement TTL option for a VIP6 address by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPs** > **IPv6s**, and enable the **Decrement TTL** parameter while adding a new VIP6 address or editing an existing address.

Configuring ARP response Suppression for Virtual IP addresses (VIPs)

September 21, 2020

You can configure the NetScaler appliance to respond or not respond to ARP requests for a Virtual IP (VIP) address on the basis of the state of the virtual servers associated with that VIP.

For example, if virtual servers V1, of type HTTP, and V2, of type HTTPs, share VIP address 10.102.29.45 on a NetScaler appliance, you can configure the appliance to not respond to any ARP request for VIP 10.102.29.45 if both V1 and V2 are in the DOWN state.

The following three options are available for configuring ARP-response suppression for a virtual IP address.

- **NONE**. The NetScaler appliance responds to any ARP request for the VIP address, irrespective of the state of the virtual servers associated with the address.
- **ONE VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if at least one of the associated virtual servers is in UP state.
- **ALL VSERVER**. The NetScaler appliance responds to any ARP request for the VIP address if all of the associated virtual servers are in UP state.

Following table shows the sample behavior of NetScaler appliance for a VIP configured with two virtual servers:

Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
NONE				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	Yes	Yes	Yes
ONE VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN

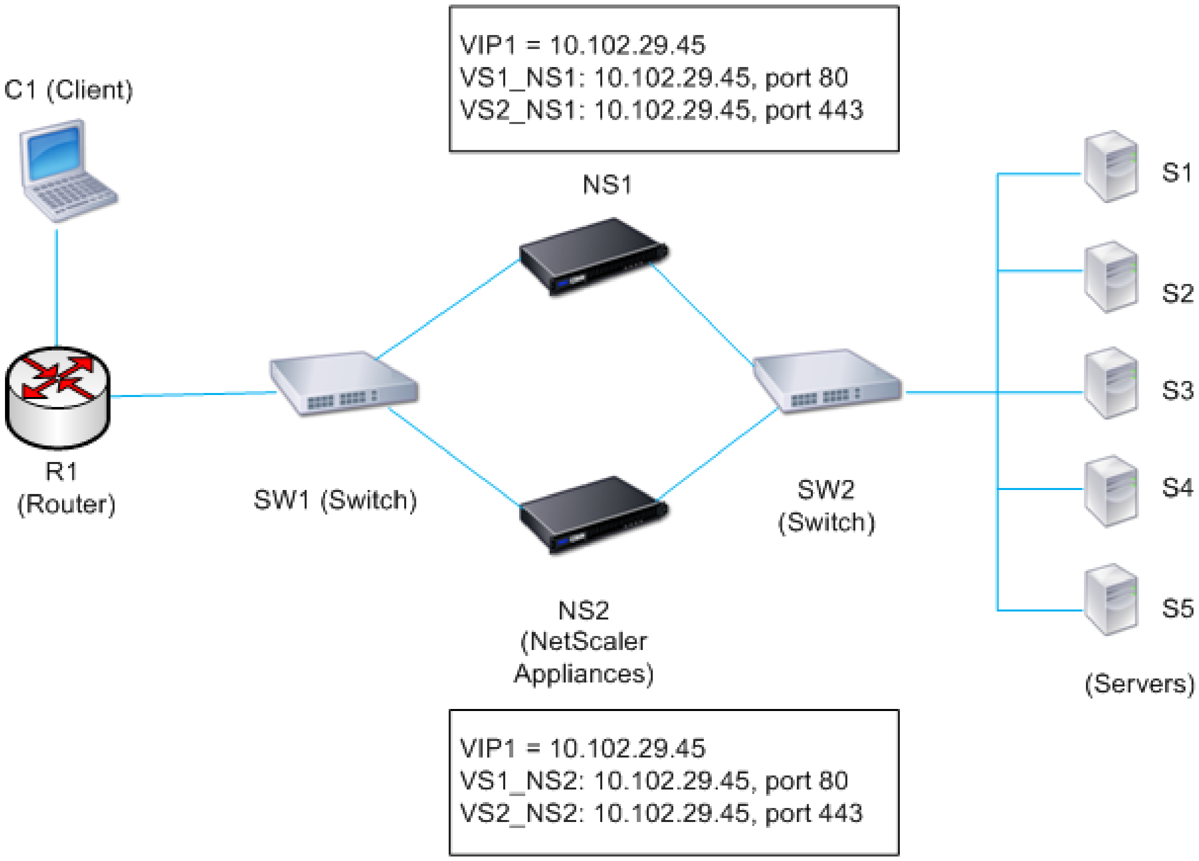
Associated virtual servers for a VIP	STATE 1	STATE 2	STATE 3	STATE 4
Respond to an ARP request for this VIP?	Yes	Yes	Yes	No
ALL VSERVER				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Respond to an ARP request for this VIP?	Yes	No	No	No

Consider an example where you want to test the performance of two virtual servers, V1 and V2, which have the same VIP address but are of different types and are each configured on NetScaler appliances NS1 and NS2. Let's call the shared VIP address *VIP1*.

V1 load balances servers S1, S2, and S3. V2 load balances servers S4 and S5.

On both NS1 and NS2, for VIP1, the ARP suppression parameter is set to ALL_VSERVER. If you want to test the performance of V1 and V2 on NS1, you must manually disable V1 and V2 on NS2, so that NS2 does not respond to any ARP request for VIP1.

Figure 1.


VIP1 = 10.102.29.45
VS1_NS1: 10.102.29.45, port 80
VS2_NS1: 10.102.29.45, port 443
C1 (Client)
NS1
S1
S2
S3
S4
S5
R1
(Router)
SW1 (Switch)
SW2
(Switch)
NS2
(NetScaler
Appliances)
(Servers)
VIP1 = 10.102.29.45
VS1_NS2: 10.102.29.45, port 80
VS2_NS2: 10.102.29.45, port 443

The execution flow is as follows:

1. Client C1 sends a request to V1. The request reaches R1.
2. R1 does not have an APR entry for the IP address (VIP1) of V1, so R1 broadcasts an ARP request for VIP1.
3. NS1 replies with source MAC address MAC1 and source IP address VIP1. NS2 does not reply to the ARP request.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table, and R1 updates the ARP entry with MAC1 and VIP1.
5. R1 forwards the packet to address VIP1 on NS1.
6. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP addresses and S2. When S2 sends a response to the client, the response returns by the same path.
7. Now you want to test the performance of V1 and V2 on NS2, so you enable V1 and V2 on NS2 and disable them on NS1. NS2 now broadcasts an ARP message for VIP1. In the message, MAC2 is the source MAC address and VIP1 is the source IP address.
8. SW1 learns the port number for reaching MAC2 from the ARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS2. R1 updates its ARP table.
9. Now suppose the ARP entry for VIP1 times out in the ARP table of R1, and client C1 sends a request for V1. Because R1 does not have an APR entry for VIP1, it broadcasts an ARP request

for VIP1.

10. NS2 replies with a source MAC address and VIP1 as the source IP address. NS1 does not reply to the ARP request.

To configure ARP response suppression by using the NetScaler command line:

At the command prompt, type:

- **set ns ip** -**arpResponse** <arpResponse>]
- **sh ns ip** <IPAddress>

Example:

```
> set ns ip 10.102.29.96 -arpResponse ALL_VSERVERS
 Done
<!--NeedCopy-->
```

To configure ARP response suppression by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Open an IP address entry and select the type of ARP Response.

Configuring Subnet IP Addresses (SNIPs)

September 21, 2020

A subnet IP address (SNIP) is a NetScaler owned IP address that is used by the NetScaler appliance to communicate with the servers.

The NetScaler appliance uses the subnet IP address as a source IP address to proxy client connections to servers. It also uses the subnet IP address when generating its own packets, such as packets related to dynamic routing protocols, or to send monitor probes to check the health of the servers. Depending on your network topology, you might have to configure one or more SNIPs for different scenarios.

To configure a SNIP address on a NetScaler appliance, you add the SNIP address and then enable global Use Subnet IP (USNIP) mode. As an alternative to creating SNIPs one at a time, you can specify a consecutive range of SNIPs.

To configure a SNIP address by using the NetScaler command line:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.203 255.255.255.0 -type SNIP
 Done
<!--NeedCopy-->
```

To create a range of SNIP addresses by using the NetScaler command line:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP
- show ns ip <IPAddress>

Example:

```
> add ns ip 10.102.29.[205-209] 255.255.255.0 -type SNIP
ip "10.102.29.205" added
ip "10.102.29.206" added
ip "10.102.29.207" added
ip "10.102.29.208" added
ip "10.102.29.209" added
 Done
<!--NeedCopy-->
```

To enable or disable USNIP mode by using the NetScaler command line:

At the command prompt, type one of the following commands:

- enable ns modeUSNIP
- disable ns modeUSNIP

To configure a SNIP address by using the NetScaler GUI:

Navigate to System > Network > IPs > IPV4s, and add a new SNIP address or edit an existing address.

To create a range of SNIP addresses by using the NetScaler GUI:

1. Navigate to System > Network > IPs > IPV4s.
2. In the Action list, select Add Range.

To enable or disable USNIP mode by using the NetScaler command line:

At the command prompt, type one of the following commands:

- enable ns mode USNIP
- disable ns mode USNIP

To enable or disable USNIP mode by using the NetScaler GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change modes.
2. Select or clear the Use Subnet IP option.

Using SNIPs for a Directly Connected Server Subnet

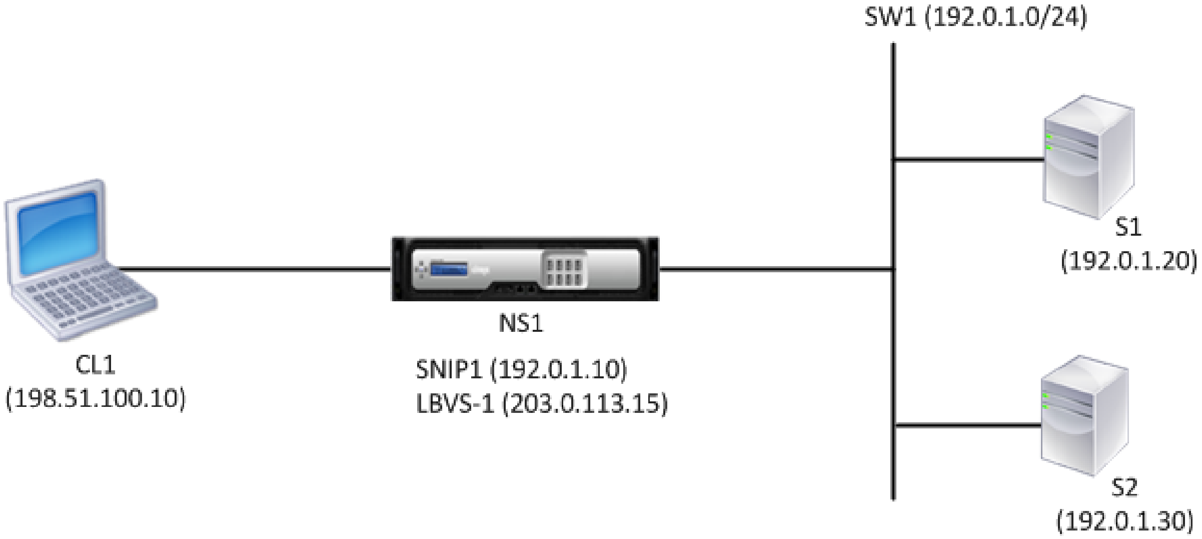
To enable communication between the NetScaler and a server that is either connected directly to the NetScaler or connected through only an L2 switch, you must configure a subnet IP address that belongs to the subnet of the server. You must configure at least one subnet IP address for each directly connected subnet, except for the directly connected management subnet that is connected through NSIP.

Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler appliance NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to the same subnet.

SNIP address SNIP1, which belongs to the same subnet as S1 and S2, is configured on NS1. As soon as SNIP1 is configured, NS1 broadcasts ARP packets for SNIP1.

Services SVC-S1 and SVC-S2 on NS1 represent S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for S1 and S2 to resolve IP-to-MAC mapping. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about configuring load balancing on a NetScaler appliance, see Load Balancing.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
    - Source IP = IP address of the client (198.51.100.10)
    - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1’s load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP1 and S2.
5. NS1 sends the request packet to S2 from SNIP1. The request packet has:

    - Source IP = SNIP1 (192.0.1.10)
    - Destination IP = IP address of S2 (192.0.1.30)
6. S2's response returns by the same path.

Using SNIPs for Server Subnets Connected through a Router

To enable communication between the NetScaler appliance and servers in subnets connected through a router, you must configure at least one subnet IP address that belongs to the subnet of the directly connected interface to the router. The appliance uses this subnet IP address to communicate with servers in subnets that can be reached through the router.

Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler appliance NS1 is used to load balance servers S1, S2, S3, and S4, which are connected to NS1 through router R1.

S1 and S2 belong to same subnet, 192.0.2.0/24, and are connected to R1 through L2 switch SW1. S3 and S4 belong to a different subnet, 192.0.3.0/24, and are connected to R1 through L2 switch SW2.

NetScaler appliance NS1 is connected to router R1 through subnet 192.0.1.0/24. SNIP address SNIP1, which belongs to the same subnet as the directly connected interface to the router (192.0.1.0/24), is configured on NS1. NS1 uses this address to communicate with servers S1 and S2, and with servers S3 and S4.

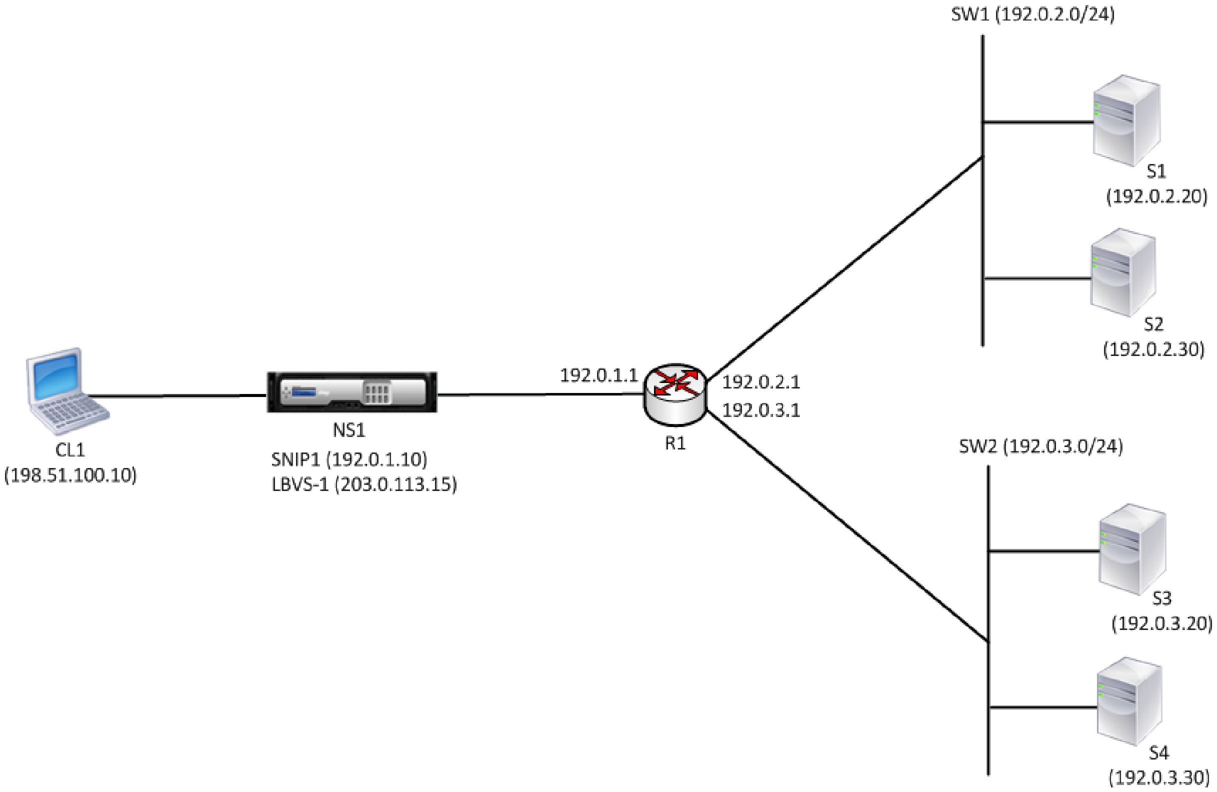
For more information about configuring load balancing on a NetScaler appliance, see Load Balancing.

As soon as address SNIP1 is configured, NS1 broadcasts ARP announcement packets for SNIP1.

NS1's routing table consists of route entries for S1, S2, S3, and S4 through R1. These route entries are either static route entries or advertised by R1 to NS1, using dynamic routing protocols.

Services SVC-S1, SVC-S2, SVC-S3, and SVC-S4 on NS1 represent servers S1, S2, S3, and S4. NS1 finds, in its routing tables, that these servers are reachable through R1. NS1 sends them monitoring probes at regular intervals, from address SNIP1, to check their health.

For more information about IP routing on a NetScaler appliance, see IP Routing.


SW1 (192.0.2.0/24)
S1
(192.0.2.20)
S2
(192.0.2.30)
192.0.1.1
192.0.2.1
192.0.3.1
R1
CL1
(198.51.100.10)
NS1
SNIP1 (192.0.1.10)
LBVS-1 (203.0.113.15)
SW2 (192.0.3.0/24)
S3
(192.0.3.20)
S4
(192.0.3.30)

Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
    - Source IP = IP address of the client (198.51.100.10)
    - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1’s load balancing algorithm selects server S3.
4. NS1 checks its routing table and finds that S3 is reachable through R1. SNIP1 (192.0.1.10) is the only IP address on NS1 that belongs to the same subnet as router R1, NS1 opens a connection between SNIP1 and S3 through R1.
5. NS1 sends the request packet to R1 from SNIP1. The request packet has:
    - Source IP address = SNIP1 (192.0.1.10)
    - Destination IP address = IP address of S3 (192.0.3.20)
6. The request reaches R1, which checks its routing table and forwards the request packet to S3.
7. S3’s response returns by the same path.

Using SNIPs for Multiple Server Subnets (VLANs) on an L2 Switch

When you have multiple server subnets (VLANs) on an L2 switch that is connected to a NetScaler appliance, you must configure at least one SNIP address for each of the server subnets, so that the NetScaler appliance can communicate with these server subnets.

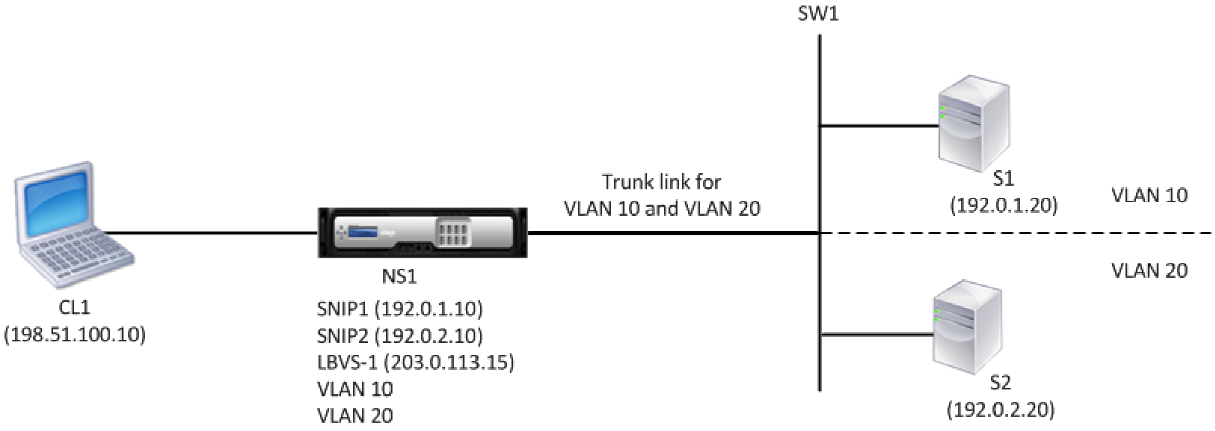
Consider an example of a load balancing setup in which load balancing virtual server LBVS1 on NetScaler appliance NS1 is used to load balance servers S1 and S2, which are connected to NS1 through L2 switch SW1. S1 and S2 belong to different subnets and are part of VLAN 10 and VLAN20, respectively. The link between NS1 and SW1 is a trunk link and is shared by VLAN10 and VLAN20.

For more information about configuring load balancing on a NetScaler appliance, see Load Balancing.

Subnet IP addresses SNIP1 (for reference purposes only) and SNIP2 (for reference purposes only) are configured on NS1. NS1 uses SNIP1 (on VLAN 10) to communicate with server S1, and SNIP2 (on VLAN 20) to communicate with S2. As soon as SNIP1 and SNIP2 are configured, NS1 broadcasts ARP announcement packets for SNIP1 and SNIP2.

For more information about configuring VLANs on a NetScaler appliance, see Configuring a VLAN.

Services SVC-S1 and SVC-S2 on NS1 represent servers S1 and S2. As soon as these services are configured, NS1 broadcasts ARP requests for them. After S1 and S2 respond, NS1 sends them monitoring probes at regular intervals to check their health. NS1 sends monitoring probes to S1 from address SNIP1, and to S2 from address SNIP2.



Following is the traffic flow in this example:

1. Client C1 sends a request packet to LBVS-1. The request packet has:
   - Source IP = IP address of the client (198.51.100.10)
   - Destination IP = IP address of LBVS-1 (203.0.113.15)
2. LBVS1 of NS1 receives the request packet.
3. LBVS1’s load balancing algorithm selects server S2.
4. Because S2 is directly connected to NS1, and SNIP2 (192.0.2.10) is the only IP address on NS1 that belongs to the same subnet as S2, NS1 opens a connection between SNIP2 and S2.
   Note: If S1 is selected, NS1 opens a connection between SNIP1 and S1.
5. NS1 sends the request packet to S2 from SNIP2. The request packet has:
   - Source IP = SNIP1 (192.0.2.10)
   - Destination IP = IP address of S2 (192.0.2.20)
6. S2’s response returns by the same path.

Configuring GSLB Site IP Addresses (GSLBIP)

September 21, 2020

A GSLB site IP (GSLBIP) address is an IP address associated with a GSLB site. It is not mandatory to specify a GSLBIP address when you initially configure the NetScaler appliance. A GSLBIP address is used only when you create a GSLB site.

For more information about creating a GSLB site IP address, see Global Server Load Balancing.

Removing a NetScaler-Owned IP Address

September 21, 2020

You can remove any IP address except the NSIP. The following table provides information about the processes you must follow to remove the various types of IP addresses. Before removing a VIP, remove the associated virtual server.

IP address type	Implications
Subnet IP address (SNIP)	If IP address being removed is the last IP address in the subnet, the associated route is deleted from the route table. If the IP address being removed is the gateway in the corresponding route entry, the gateway for that subnet route is changed to another NetScaler-owned IP address.
Virtual Server IP address (VIP)	Before removing a VIP, you must first remove the virtual server associated with it. For information about removing the virtual server, see Load Balancing.
GSLB-Site-IP address	Before removing a GSLB site IP address, you must remove the site associated with it. For information about removing the site, see Global Server Load Balancing.

To remove an IP address by using the NetScaler command line:

At the command prompt, type:

rm ns ip <IPaddress>

Example:

```
> rm ns ip 10.102.29.54
Done
<!--NeedCopy-->
```

To remove an IP address by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPs** > **IPV4s**, delete the IP address.

Configuring Application Access Controls

September 21, 2020

Application access controls, also known as management access controls, form a unified mechanism for managing user authentication and implementing rules that determine user access to applications and data. You can configure SNIPs to provide access for management applications. Management access for the NSIP is enabled by default and cannot be disabled. You can, however, control it by using ACLs.

For information about using ACLs, see Access Control Lists (ACLs).

The NetScaler appliance does not support management access to VIPs.

The following table provides a summary of the interaction between management access and specific service settings for Telnet.

Management Access	Telnet (State Configured on the NetScaler)	Telnet (Effective State at the IP Level)
Enable	Enable	Enable
Enable	Disable	Disable
Disable	Enable	Disable
Disable	Disable	Disable

The following table provides an overview of the IP addresses used as source IP addresses in outbound traffic.

Application/ IP	NSIP	SNIP	VIP
ARP	Yes	Yes	No
Server side traffic	No	Yes	No
RNAT	No	Yes	Yes
ICMP PING	Yes	Yes	No
Dynamic routing	Yes	Yes	Yes

The following table provides an overview of the applications available on these IP addresses.

Application/ IP	NSIP	SNIP	VIP
SNMP	Yes	Yes	Yes
System access	Yes	Yes	No

You can access and manage the NetScaler by using applications such as Telnet, SSH, GUI, and FTP.

Note: Telnet and FTP are disabled on the NetScaler for security reasons. To enable them, contact the customer support. After the applications are enabled, you can apply the controls at the IP level.

To configure the NetScaler to respond to these applications, you need to enable the specific management applications. If you disable management access for an IP address, existing connections that use the IP address are not terminated, but no new connections can be initiated.

Also, the non-management applications running on the underlying FreeBSD operating system are open to protocol attacks, and these applications do not take advantage of the NetScaler appliance's attack prevention capabilities.

You can block access to these non-management applications on a SNIP or NSIP. When access is blocked, a user connecting to a NetScaler by using the SNIP or NSIP is not be able to access the non-management applications running on the underlying operating system.

To configure management access for an IP address by using the NetScaler command line:

At the command prompt, type:

set ns ip <IPAddress> -mgmtAccess <value> -**telnet** <value> -**ftp** <value> -**gui** <value> -**ssh** <value> -**snmp** <value> -**restrictAccess** (**ENABLED** | **DISABLED**)

Example:

```
To enable management access and restrict access for a NetScaler owned
   IP address:
```

```

 > set ns ip 192.0.2.20 -mgmtAccess enabled -restrictAccess ENABLED
  Done

To disable GUI access for a NSIP or SNIP address:

 > set ns ip 192.0.2.10 -gui disabled
  Done

<!--NeedCopy-->
```

To enable management access for an IP address by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Open an IP address entry, and select the **Enable Management Access control** to support the listed applications option.

How the NetScaler Proxies Connections

September 21, 2020

When a client initiates a connection, the NetScaler appliance terminates the client connection, initiates a connection to an appropriate server, and sends the packet to the server. The appliance does not perform this action for service type UDP or ANY.

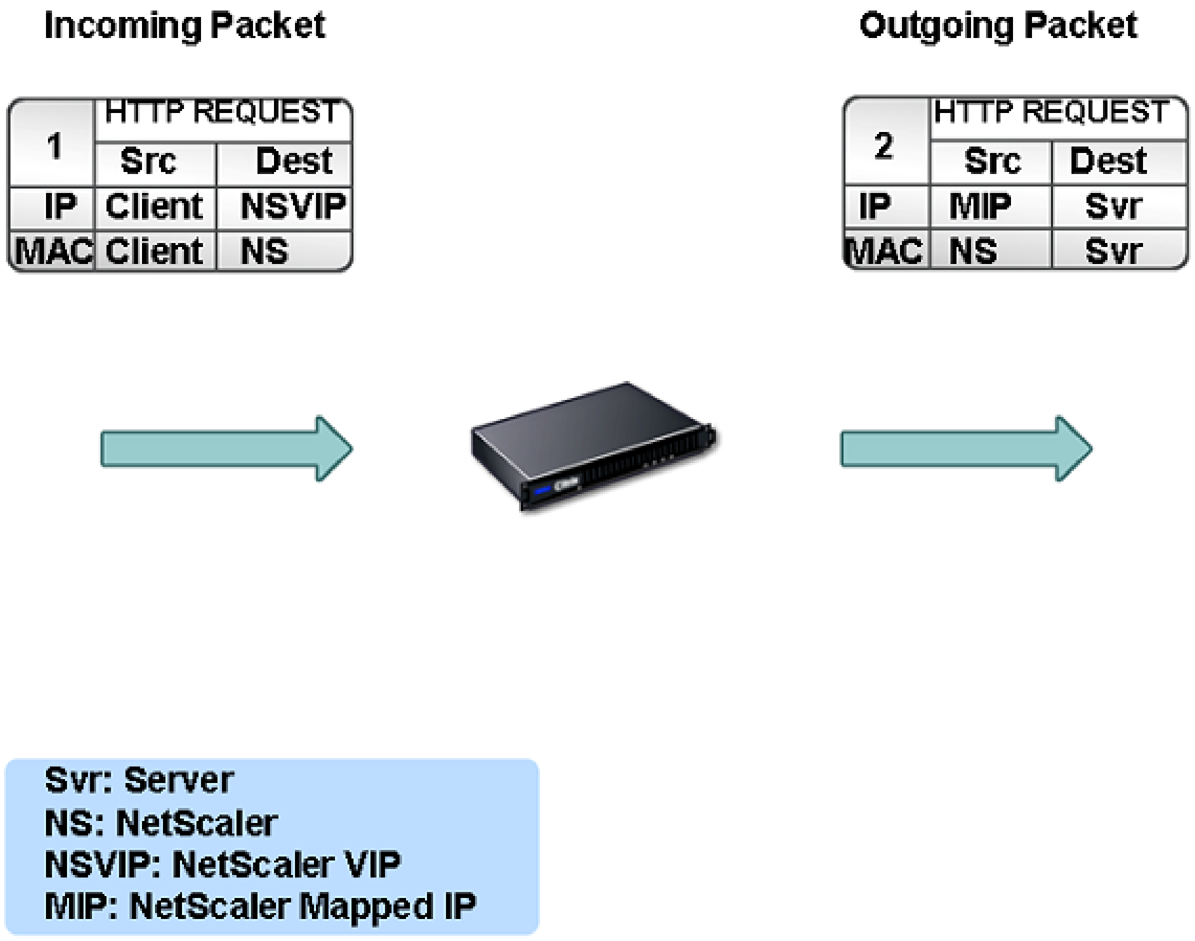
For more information about service types, see Load Balancing.

You can configure the NetScaler to process the packet before initiating the connection with a server. The default behavior is to change the source and destination IP addresses of a packet before sending the packet to the server. You can configure the NetScaler to retain the source IP address of the packets by enabling Use Source IP mode.

How the Destination IP Address Is Selected

Traffic sent to the NetScaler appliance can be sent to a virtual server or to a service. The appliance handles traffic to virtual servers and services differently. The NetScaler terminates traffic received at a virtual server IP (VIP) address and changes the destination IP address to the IP address of the server before forwarding the traffic to the server, as shown in the following diagram.

Figure 1. Proxying Connections to VIPs



Incoming Packet

1	HTTP REQUEST	
	Src	Dest
IP	Client	NSVIP
MAC	Client	NS

Outgoing Packet

2	HTTP REQUEST	
	Src	Dest
IP	MIP	Svr
MAC	NS	Svr

Svr: Server
NS: NetScaler
NSVIP: NetScaler VIP
MIP: NetScaler Mapped IP

Packets destined for a service are sent directly to the appropriate server, and the NetScaler does not modify the destination IP addresses. In this case, the NetScaler functions as a proxy.

How the Source IP Address Is Selected

When the NetScaler appliance communicates with the physical servers or peer devices, by default, it does not use the IP address of the client. NetScaler maintains a pool of subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address of a connection to the physical server. Depending on the subnet in which the physical server is placed, NetScaler selects a specific SNIP address.

Note: If the Use Source IP (USIP) option is enabled, NetScaler uses the IP address of the client.

Enable Use Source IP Mode

September 21, 2020

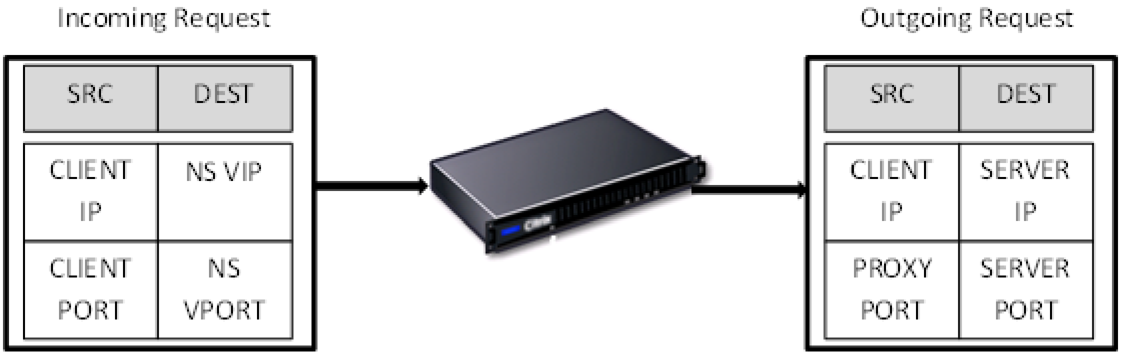
When the NetScaler appliance communicates with the physical servers or peer devices, by default, it uses one of its own IP addresses as the source IP. The appliance maintains a pool of subnet IP addresses (SNIPs), and selects an IP address from this pool to use as the source IP address for a connection to the physical server. The decision of selecting a SNIP address depends on the subnet in which the physical server resides.

If necessary, you can configure the NetScaler appliance to use the client's IP address as source IP. Some applications need the actual IP address of the client. The following use cases are a few examples:

- Client's IP address in the web access log is used for billing purposes or usage analysis.
- Client's IP address is used to determine the country of origin of the client or the originating ISP of the client. For example, many search engines such as Google provide content relevant to the location to which the user belongs.
- The application must know the client's IP address to verify that the request is from a trustworthy source.
- Sometimes, even though an application server does not need the client's IP address, a firewall placed between the application server and the NetScaler may need the client's IP address for filtering the traffic.

Enable Use Source IP mode (USIP) mode if you want the NetScaler to use the client's IP address for communication with the servers.

The following figure shows how the NetScaler uses IP addresses in USIP mode.



Incoming Request

SRC	DEST
CLIENT IP	NS VIP
CLIENT PORT	NS VPORT

Outgoing Request

SRC	DEST
CLIENT IP	SERVER IP
PROXY PORT	SERVER PORT

Before you begin

Before you enable USIP mode, note the following points:

- Enable USIP in the following situations:
  - Load balancing of Intrusion Detection System (IDS) servers
  - SMTP load balancing
  - Stateless connection failover
  - Sessionless load balancing
  - If you use the Direct Server Return (DSR) mode
- The USIP global setting applies only to services that are created after the USIP global setting is made. In other words, the USIP global setting does not apply to the existing services when the USIP global setting is made. For example, Disabling USIP globally does not disable USIP on the existing services. But it stops the subsequently created services from having USIP enabled automatically.

  To enable or disable USIP on a set of existing services, you need to enable or disable USIP on each of these services.
- When USIP is enabled, you must set server's gateway to one of the NetScaler owned IP addresses (of type Subnet IP (SNIP) so that server's response always go through the NetScaler appliance.
- If you enable USIP, set the idle timeout for server connections to a value lower than the default value, so that idle connections are cleared quickly on the server side.
- For transparent cache redirection, if you enable USIP, enable L2CONN also.
- Because HTTP connections are not reused when USIP is enabled, a large number of server-side connections may accumulate. Idle server connections can block connections for other clients. Therefore, set limits on maximum number of connections to a service. Citrix also recommends setting the HTTP server time-out value, for a service on which USIP is enabled, to a value lower than the default, so that idle connections are cleared quickly on the server side.
- As an alternative to USIP mode, you have the option of inserting the client's IP address (CIP) in the request header of the server-side connection for an application server that needs the client's IP address.
- In earlier NetScaler releases, USIP mode had the following source-port options for server-side connections:
  - **Use the client's port**. With this option, connections cannot be reused. For every request from the client, a new connection is made with the physical server.
  - **Use proxy port**. With this option, connection reuse is possible for all requests from the same client.

In the later NetScaler releases, if USIP is enabled, the default is to use a proxy port for server-side connections and not reuse connections. Not reusing connections may not affect the speed of establishing connections.

By default, the Use Proxy Port option is enabled if the USIP mode is enabled.

Note: If you enable the USIP mode, it is recommended to enable the Use Proxy Port option.

For more information about the Use Proxy Port option, see Configure the source port for server-side connections.

Configuration Steps

Enable Use Source IP mode (USIP) mode if you want NetScaler to use the client's IP address for communication with the servers. By default, USIP mode is disabled. USIP mode can be enabled globally on the NetScaler or on a specific service. If you enable it globally, USIP is enabled by default for all subsequently created services. If you enable USIP for a specific service, the client's IP address is used only for the traffic directed to that service.

NetScaler command line procedures

To globally enable or disable USIP mode by using the NetScaler command line:

At the command prompt, type one of the following commands:

- **enable ns mode USIP**
- **disable ns mode USIP**

To enable USIP mode for a service by using the NetScaler command line:

At the command prompt, type:

set service <name>@ -**usip** (**YES** | **NO**)

Example:

```
> set service Service-HTTP-1 -usip YES
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To globally enable or disable USIP mode by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change Modes**.
2. Select or clear the **Use Source IP** option.

To enable USIP mode for a service by using the NetScaler GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and edit a service.
2. In **Advanced Settings**, select **Service Settings**, and select **Use Source IP Address**.

Configuring Network Address Translation

September 21, 2020

Network address translation (NAT) involves modification of the source and/or destination IP addresses and/or the TCP/UDP port numbers of IP packets that pass through the NetScaler appliance. Enabling NAT on the appliance enhances the security of your private network, and protects it from a public network such as the Internet, by modifying your networks source IP addresses when data passes through the NetScaler. Also, with the help of NAT entries, your entire private network can be represented by a few shared public IP addresses. The NetScaler supports the following types of network address translation:

- **Inbound NAT (INAT).** The NetScaler replaces the destination IP address in the packets generated by the client with the private IP address of the server.
- **Reverse NAT (RNAT).** The NetScaler replaces the source IP address in the packets generated by the servers with the public NAT IP addresses.

INAT

September 21, 2020

When a client sends a packet to a NetScaler appliance that is configured for Inbound Network Address Translation (INAT), the appliance translates the packet's public destination IP address to a private destination IP address and forwards the packet to the server at that address.

The following configurations are supported:

- **IPv4-IPv4 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.
- **IPv4-IPv6 Mapping**: A public IPv4 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance creates an IPv6 request packet with the IP address of the IPv6 server as the destination IP address.

- **IPv6-IPv4 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv4 server. The NetScaler appliance creates an IPv4 request packet with the IP address of the IPv4 server as the destination IP address.
- **IPv6-IPv6 Mapping**: A public IPv6 address on the NetScaler appliance listens to connection requests on behalf of a private IPv6 server. The NetScaler appliance translates the packet's public destination IP address to the destination IP address of the server and forwards the packet to the server at that address.

When the appliance forwards a packet to a server, the source IP address assigned to the packet is determined as follows:

- If use subnet IP (USNIP) mode is enabled and use source IP (USIP) mode is disabled, the NetScaler uses a subnet IP address (SNIP) as the source IP address.
- If USNIP mode is disabled and USIP mode is disabled, the NetScaler uses a mapped IP address (MIP) as the source IP address.
- If USIP mode is enabled, and USNIP mode is disabled the NetScaler uses the client IP (CIP) address as the source IP address.
- If both USIP and USNIP modes are enabled, USIP mode takes precedence.
- You can also configure the NetScaler to use a unique IP address as the source IP address, by setting the proxyIP parameter.
- If none of the above modes are enabled and a unique IP address has not been specified, the NetScaler attempts to use a MIP as the source IP address.
- If both USIP and USNIP modes are enabled and a unique IP address has been specified, the order of precedence is as follows: USIP-unique IP-USNIP-MIP-Error.

To protect the NetScaler from DoS attacks, you can enable TCP proxy. However, if other protection mechanisms are used in your network, you may want to disable them.

Configure INAT

You can create, modify, or remove an INAT entry.

NetScaler command line procedures

To create an INAT entry by using the NetScaler command line:

At the command prompt, type the following commands to create an INAT entry and verify its configuration:

- **add inat** <name> <publicIP> <privateIP> [-**tcpproxy** (**ENABLED** | **DISABLED**)] [-**ftp** (**ENABLED** | **DISABLED**)] [-**usip** (**ON** | **OFF**)] [-**usnip** (**ON** | **OFF**)] [-**proxyIP** <ip_addr|ipv6_addr>]
- **show inat** [<name>]

Example:

```
> add inat ip4-ip4 172.16.1.2 192.168.1.1 -proxyip 10.102.29.171
 Done
<!--NeedCopy-->
```

To modify an INAT entry by using the NetScaler command line:

To modify an INAT entry, type the **set inat **command, the name of the entry, and the parameters to be changed, with their new values.

To remove an INAT configuration by using the NetScaler command line:

At the command prompt, type:

- **rm inat** <name>

Example:

```
> rm inat ip4-ip4
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To configure an INAT entry by using the NetScaler GUI:

Navigate to **System** > **Network** > **Routes** > **INAT**, and add a new INAT entry or edit an existing INAT entry.

To remove an INAT configuration by using the NetScaler GUI:

Navigate to **System** > **Network** > **Routes** > **INAT**, delete the INAT configuration.

Coexistence of INAT and Virtual Servers

September 21, 2020

If both INAT and RNAT are configured, the INAT rule takes precedence over the RNAT rule. If RNAT is configured with a network address translation IP (NAT IP) address, the NAT IP address is selected as the source IP address for that RNAT client.

The default public destination IP in an INAT configuration is the virtual IP (VIP) address of the NetScaler device. Virtual servers also use VIPs. When both INAT and a virtual server use the same IP address, the Vserver configuration overrides the INAT configuration.

Following are a few sample configuration setup scenarios and their effects.

Case	Result
You have configured a virtual server and a service to send all data packets received on a specific NetScaler port to the server directly. You have also configured INAT and enabled TCP. Configuring INAT in this manner sends all data packets received through a TCP engine before sending them to the server.	All packets received on the NetScaler, except those received on the specified port, pass through the TCP engine.
You have configured a virtual server and a service to send all data packets of service type TCP, that are received on a specific port on the NetScaler, to the server after passing through the TCP engine. You have also configured INAT and disabled TCP. Configuring INAT in this manner sends the data packets received directly to the server.	Only packets received on the specified port pass through the TCP engine.
You have configured a virtual server and a service to send all data packets received to either of two servers. You are attempting to configure INAT to send all data packets received to a different server.	The INAT configuration is not allowed.
You have configured INAT to send all received data packets directly to a server. You are attempting to configure a virtual server and a service to send all data packets received to two different servers.	The vserver configuration is not allowed.

Stateless NAT46

September 21, 2020

The stateless NAT46 feature enables communication between IPv4 and IPv6 networks through IPv4 to IPv6 packet translation, and vice versa, without maintaining any session information on the NetScaler appliance.

For a stateless NAT46 configuration, the appliance translates an IPv4 packet to IPv6 or an IPv6 packet to IPv4 as defined in RFCs 6145 and 2765.

A stateless NAT46 configuration on the NetScaler appliance has the following components:

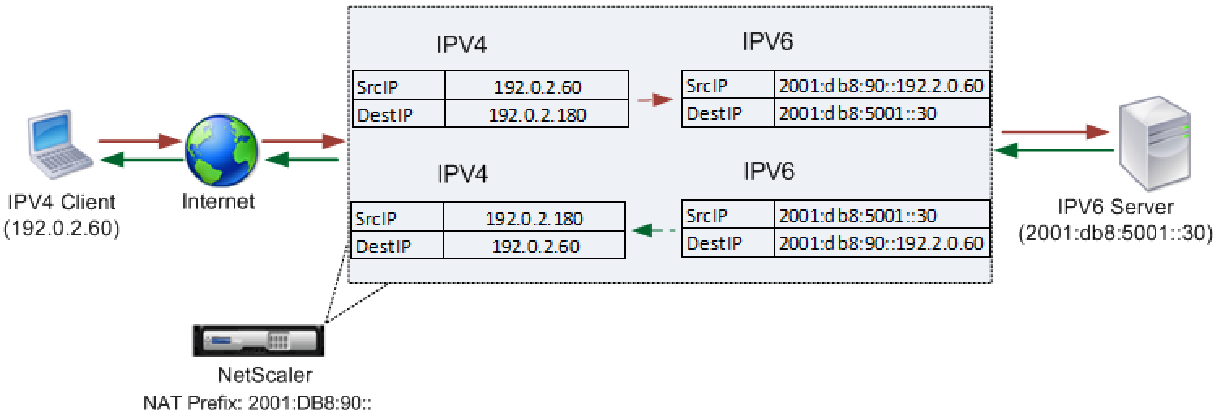
- **IPv4-IPv6 INAT entry.** An INAT entry defining a 1:1 relationship between an IPv4 address and an IPv6 address. In other words, an IPv4 address on the appliance listens to connection requests on behalf of an IPv6 server. An IPv4 request packet for this IPv4 address is translated into an IPv6 packet, and then the IPv6 packet is sent to the IPv6 server.

  The appliance translates an IPv6 response packet into an IPv4 response packet with its source IP address field set as the IPv4 address specified in the INAT entry. The translated packet is then sent to the client.

- **NAT46 IPv6 prefix.** A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance. During IPv4 packet to IPv6 packet translation, the appliance sets the source IP address of the translated IPv6 packet to a concatenation of the NAT46 IPv6 prefix [96 bits] and the IPv4 source address [32 bits] that was received in the request packet.

  During IPv6 packet to IPv4 packet translation, the appliance sets the destination IP address of the translated IPv4 packet to the last 32 bits of the destination IP address of the IPv6 packet.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv6 address. To enable communication between IPv4 clients and IPv6 server S1, NetScaler appliance NS1 is deployed with a stateless NAT46 configuration that includes an IPv4-IPv6 INAT entry for server S1, and a NAT46 Prefix. The INAT entry includes an IPv4 address at which the appliance listens to connection requests from IPv4 clients on behalf of the IPv6 server S1.


IPV4
SrcIP 192.0.2.60
DestIP 192.0.2.180
IPV6
SrcIP 2001:db8:90::192.2.0.60
DestIP 2001:db8:5001::30
IPV4
SrcIP 192.0.2.180
DestIP 192.0.2.60
IPV6
SrcIP 2001:db8:5001::30
DestIP 2001:db8:90::192.2.0.60
IPV4 Client
(192.0.2.60)
Internet
IPV6 Server
(2001:db8:5001::30)
NetScaler
NAT Prefix: 2001:DB8:90::

The following table lists the settings used in this example:

Entities	Name	Value
IP address of the client	Client_IPv4 (for reference purposes only)	192.0.2.60
IPv6 address of the server	Sevr_IPv6 (for reference purposes only)	2001:DB8:5001::30

Entities	Name	Value
IPv4 address defined in the INAT entry for IPv6 server S1	Map-Sevr-IPv4 (for reference purposes only)	192.0.2.180
IPv6 prefix for NAT 46 translation	NAT46_Prefix (for reference purposes only)	2001:DB8:90::

Following is the traffic flow in this example:

1. IPv4 Client CL1 sends a request packet to the Map-Sevr-IPv4 (192.0.2.180) address on the NetScaler appliance.
2. The appliance receives the request packet and searches the NAT46 INAT entries for the IPv6 address mapped to the Map-sevr-IPv4 (192.0.2.180) address. It finds the Sevr-IPv6 (2001:DB8:5001::30) address.
3. The appliance creates a translated IPv6 request packet with:
    - Destination IP address field = Sevr-IPv6 = 2001:DB8:5001::30
    - Source IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits) = 2001:DB8:90::192.0.2.60
4. The appliance sends the translated IPv6 request to Sevr-IPv6.
5. The IPv6 server S1 responds by sending an IPv6 packet to the NetScaler appliance with:
    - Destination IP address field = Concatenation of NAT Prefix (First 96 bits) and Client_IPv4 (last 32 bits)= 2001:DB8:90::192.0.2.60
    - Source IP address field = Sevr-IPv6 = 2001:DB8:5001::30
6. The appliance receives the IPv6 response packet and verifies that its destination IP address matches the NAT46 prefix configured on the appliance. Because the destination address matches the NAT46 prefix, the appliance searches the NAT46 INAT entries for the IPv4 address associated with the Sevr-IPv6 address (2001:DB8:5001::30). It finds the Map-Sevr-IPv4 address (192.0.2.180).
7. The appliance creates an IPv4 response packet with:
    - Destination IP address field = The NAT46 prefix stripped from the destination address of the IPv6 response = Client_IPv4 (192.0.2.60)
    - Source IP address field = Map-Sevr-IPv4 address (192.0.2.180)
8. The appliance sends the translated IPv4 response to client CL1.

Limitations of Stateless NAT46

The following limitations apply to stateless NAT46:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.

- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv4 packets is not supported.
- Translation of multicast packets is not supported.
- Translation of destination option headers and source routing headers is not supported.
- Translation of fragmented IPv4 UDP packets that do not contain UDP checksum is not supported.

Configure Stateless NAT46

Creating the required entities for stateless NAT46 configuration on the NetScaler appliance involves the following procedures:

1. Create an IPv4-IPv6 mapping INAT entry with stateless mode enabled.
2. Create a NAT46 IPv6 prefix.

NetScaler command line procedures

To configure an INAT mapping entry by using the NetScaler command line:

At the command prompt, type:

- add inat <name> <publicIPv4> <privateIPv6> -mode STATELESS
- show inat <name>

To create an NAT46 prefix by using the NetScaler command line:

At the command prompt, type:

```
set inatparam -nat46v6Prefix <ipv6_addr            *>
```

-
- show inatparam

Example:

```
 > add inat exmpl-com-stls-nat46 192.0.2.180
2001:DB8:5001::30 -mode stateless
Done

> set inatparam -nat46v6Prefix 2001:DB8:90::/96
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To create an INAT mapping entry by using the NetScaler GUI:

1. Navigate to System > Network > Routes > INAT.
2. Add a new INAT entry, or edit an existing INAT entry.
3. Set the following parameters:
   - Name*
   - Public IP Address*
   - Private IP Address* (Select the IPv6 check box and enter the address in IPv6 format.)
   - Mode (Select Stateless from the drop down list.)

   * A required parameter

To create a NAT46 prefix by using the NetScaler GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure INAT Parameters**, and set the **Prefix** parameter.

Setting Global Parameters for Stateless NAT46

The appliance provides some optional global parameters for stateless NAT46 configurations.

To set global parameters for stateless NAT46 by using the NetScaler command line:

At the command prompt, type:

set inatparam [-**nat46IgnoreTOS** (**YES**	**NO**)] [-**nat46ZeroCheckSum** (**ENABLED**	**DISABLED**)] [-**nat46v6Mtu** <positive_integer>] [-**nat46FragHeader** (**ENABLED**	**DISABLED**)]

-
- **show inatparam**

Example:

```
> set inatparam -nat46IgnoreTOS YES -nat46ZeroCheckSum DISABLED -nat46v6Mtu 1400 -nat46FragHeader DISABLED
 Done
<!--NeedCopy-->
```

To set global parameters for stateless NAT46 by using the NetScaler GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure INAT Parameters**.

DNS64

September 21, 2020

The NetScaler DNS64 feature responds with a synthesized DNS AAAA record to an IPv6 client sending an AAAA request for an IPv4-only domain. The DNS64 feature is used with the NAT64 feature to enable seamless communication between IPv6-only clients and IPv4-only servers. DNS64 enables discovery of the IPv4domain by the IPV6 only clients, and NAT64 enables communication between the clients and servers.

For synthesizing an AAAA record, the NetScaler appliance fetches a DNS A record from a DNS server. The DNS64 prefix is a 96-bit IPv6 prefix configured on the NetScaler appliance. The NetScaler appliance synthesizes the AAAA record by concatenation of the DNS64 Prefix (96 bits) and the IPv4 address (32 bits).

For enabling communication between IPv6 clients and IPv4 servers, a NetScaler appliance with DNS64 and NAT64 configuration can deployed either on the IPv6 client side or on the IPv4 server side. In both cases, the DNS64 configuration on the NetScaler appliance is similar and includes a load balancing virtual server acting as a proxy server for DNS servers. If the NetScaler appliance is deployed on the client side, the load balancing virtual server must be specified, on the IPv6 client, as the nameserver for a domain.

Consider an example where a NetScaler appliance with DNS64 and NAT64 configuration is configured on the IPv4 side. In this example, an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, NetScaler appliance NS1 is deployed with a DNS64 and stateful NAT64 configuration.

The DNS64 configuration includes DNS load balancing virtual server LBVS-DNS64-1, on which the DNS64 option is enabled. A DNS64 policy named DNS64-Policy-1, and an associated DNS64 action named DNS64-Action-1, are also configured on NS1, andDNS64-Policy-1 is bound to LBVS-DNS64-1. LBVS-DNS64-1 acts as a DNS proxy server for DNS servers DNS-1 and DNS-2.

When traffic arriving at LBVS-DNS64-1 matches the conditions specified in DNS64-Policy-1, the traffic is processed according to the settings in DNS64-Action-1. DNS64-Action-1 specifies the DNS64 prefix used, with the A record received from a DNS server, to synthesize an AAAA record.

The global DNS parameter cacherecords is enabled on the NetScaler appliance, so the appliance caches DNS records. This setting is necessary for the DNS64 to work properly.

The following table lists the settings used in the above example: DNS64 example settings.

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a DNS AAAA request for the IPv6 address of the site www.example.com.
2. The request is received by the DNS load balancing virtual server LBVS-DNS64-1 on NetScaler appliance NS1.
3. NS1 checks its DNS cache records for the requested AAAA record and finds that AAAA record for the site www.example.com does not exist in the DNS cache.
4. LBVS-DNS64-1's load balancing algorithm selects DNS server DNS-1 and forwards the AAAA request to it.
5. Because the site www.example.com is hosted on an IPv4 server, the DNS server DNS-1 does not have any AAAA record for the site www.example.com.
6. DNS-1 sends either an empty DNS AAAA response or an error message to LBVS-DNS64-1.
7. Because DNS64 option is enabled on LBVS-DNS64-1 and the AAAA request from CL1 matches the condition specified in DNS64-Policy-1, NS1 sends a DNS A request to DNS-1 for the IPv4 address of www.example.com.
8. DNS-1 responds by sending the DNS A record for www.example.com to LBVS-DNS64-1. The A record includes the IPv4 address for www.example.com.
9. NS1 synthesizes an AAAA record for the site www.example.com with:
   - IPv6 address for site www.example.com = Concatenation of DNS64 Prefix (96 bits) specified in the associated DNS64action, and IPv4 address of DNS A record (32 bits) = 2001:DB8:300::192.0.2.60
10. NS1 sends the synthesized AAAA record to IPv6 client CL1. NS1 also caches the A record into its memory. NS1 uses the cached A record to synthesize AAAA records for subsequent AAAA requests.

Points to Consider for a DNS64 Configuration

Before configuring DNS64 on a NetScaler appliance, consider the following points:

- The DNS64 feature of the NetScaler appliance is compliant with RFC 6174.
- The DNS64 feature of the NetScaler appliance does not support DNSSEC. The NetScaler appliance does not synthesize an AAAA record from a DNSSEC response received from a DNS server. A response is classified as a DNSSEC response, only if it contains RRSIG records.
- The NetScaler appliance supports DNS64 prefix of length of only 96 bits.
- Though the DNS64 feature is used with the NAT64 feature, the DNS64 and NAT64 configurations are independent on the NetScaler appliance. For a particular flow, you must specify the same IPv6 prefix value for the DNS64 prefix and the NAT64 prefix parameters, so that the synthesized IPv6 addresses received by the client are routed to the particular NAT64 configuration. For more information on configuring NAT64 on a NetScaler appliance, see Stateful NAT64.
- The following are the different cases of DN64 processing by the NetScaler appliance:

  - If the AAAA response from the DNS server includes AAAA records, then each record in the response is checked for the set of exclusion rule configured on the NetScaler appliance for the particular DNS64 configuration. The NetScaler removes the IPv6 addresses, whose prefix matches the exclusion rule, from the response. If the resulting response includes at least one IPv6 record, the NetScaler appliance forwards this response to the client, else, the appliance synthesizes a AAAA response from the A record of the domain and sends it to the IPv6 client.
  - If the AAAA response from the DNS server is an empty answer response, the appliance requests for A resource records with the same domain name or searches in its own records if the appliance is an authentic domain name server for the domain. If the request results in an empty answer or error, the same is forwarded to the client.
  - If the response from the DNS server includes RCODE=1 (format error), the NetScaler appliance forwards the same to the client. If there is no response before the timeout, the NetScaler appliance sends a response with RCODE=2 (server failure) to the client.
  - If the response from the DNS server includes a CNAME, the chain is followed until the terminating A or AAAA record is reached. If the CNAME does not have any AAAA resource records, the NetScaler appliance fetches the DNS A record to be used for synthesizing AAAA record. The CNAME chain is added to the answer section along with the synthesized AAAA record and then sent to the client.
- The DNS64 feature of the NetScaler appliance also supports responding to PTR request. When a PTR request for a domain of an IPv6 address is received on the appliance and the IPv6 address matches any of the configured DNS64 prefix, the appliance creates a CNAME record mapping the IP6-ARPA domain into the corresponding IN-ADDR. ARPA domain and the newly formed IN-ADDR.ARPA domain is used for resolution. The appliance searches the local PTR records and if the records are not present, the appliance sends a PTR request for IN-ADDR.ARPA domain to the DNS server. The NetScaler appliance uses the response from the DNS server to synthesize response for the initial PTR request.

Configuration Steps

Creating the required entities for stateful NAT64 configuration on the NetScaler appliance involves the following procedures:

- **Add DNS services.** DNS services are logical representation of DNS servers for which the NetScaler appliance acts as a DNS proxy server. For more information on setting optional parameters of a service, see Load Balancing.
- **Add DNS64 action and DNS64 policy and then bind the DNS64 action to the DNS64 policy.** A DNS64 policy specifies conditions to be matched against traffic for DNS64 processing accord-

ing to the settings in the associated DNS64 action. The DNS64 action specifies the mandatory DNS64 prefix and the optional exclude rule and mapped rule settings.

- **Create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it.** The DNS load balancing virtual server acts as a DNS proxy server for DNS servers represented by the bound DNS services. Traffic arriving at the virtual server is matched against the bound DNS64 policy for DNS64 processing. For more information on setting optional parameters of a load balancing virtual server, see Load Balancing.

  Note: The NetScaler command line has separate commands for these two tasks, but the NetScaler GUI combines them in a single dialog box.

  Enable caching of DNS records. Enable the global parameter for the NetScaler appliance to cache DNS records, which are obtained through DNS proxy operations. For more information on enabling caching of DNS records, see Domain Name System.

  </div>

NetScaler command line procedures

To create a service of type DNS by using the NetScaler command line:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> …

To create a DNS64 action by using the NetScaler command line:

At the command prompt, type:

- add dns action64 <actionName> -Prefix <ipv6_addr|*> [-mappedRule <expression>] [-excludeRule <expression>]

To create a DNS64 policy by using the NetScaler command line:

At the command prompt, type:

- add dns policy64 <name> -rule <expression> -action <string>

To create a DNS load balancing virtual server by using the NetScaler command line:

At the command prompt, type:

- add lb vserver <name> DNS <IPAddress> <port> -dns64 (ENABLED | DISABLED) [-bypassAAAA (YES | NO)] …

To bind the DNS services and the DNS64 policy to the DNS load balancing virtual server by using the NetScaler command line:

At the command prompt, type:

- bind lb vserver <name> <serviceName> …
- bind lb vserver <name> -policyName <string> -priority <positive_integer> …

NetScaler GUI procedures

To create a service of type DNS by using the NetScaler GUI:

1. Navigate to Traffic Management > Load Balancing > Services, and add a new service.
2. Set the following parameters:
    - Service Name*
    - Server*
    - Protocol* (Select DNS from the drop down list.)
    - Port*

To create a DNS64 action by using the NetScaler GUI:

Navigate to Traffic Management > DNS > Actions, on the DNS Actions64 tab, add a new DNS64 action.

To create a DNS64 policy by using the NetScaler GUI:

Navigate to Traffic Management > DNS > Policies, on the DNS Policies64 tab, add a new DNS64 policy.

To create a DNS load balancing virtual server and bind the DNS services and the DNS64 policy to it by using the NetScaler GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and add a new virtual server.
2. Set the following parameters:
    - Name*
    - IP Address*
    - Protocol* (Select DNS from the drop down list.)
    - Port*
3. Select the Enable DNS64 option.
4. In the Services pane, bind the service to the virtual server.
5. In the Policies pane, bind the policy to the virtual server.

Sample Configuration

```
 > add service SVC-DNS-1 203.0.113.50 DNS 53
 Done

> add service SVC-DNS-2 203.0.113.60 DNS 53
 Done

> add dns Action64 DNS64-Action-1 -Prefix 2001:DB8:300::/96
 Done
```

```

> add dns Policy64 DNS64-Policy-1 -rule "CLIENT.IPv6.SRC.IN_SUBNET(2001:DB8:5001::/64)"
-action DNS64-Action-1
 Done

> add lb vserver LBVS-DNS64-1 DNS 2001:DB8:9999::99 53 -dns64 ENABLED
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-1
 Done

> bind lb vserver LBVS-DNS64-1 SVC-DNS-2
 Done

> bind lb vserver LBVS-DNS64-1 -policyname DNS64-Policy-1 -priority 2
 Done

<!--NeedCopy-->
```

Stateful NAT64 Translation

September 21, 2020

The stateful NAT64 feature enables communication between IPv6 clients and IPv4 servers through IPv6 to IPv4 packet translation, and vice versa, while maintaining session information on the NetScaler appliance.

A stateful NAT64 configuration on the NetScaler appliance has the following components:

- **NAT64 rule**— An entry consisting of an ACL6 rule and a netprofile, which consists of a pool of NetScaler owned SNIP Addresses.
- **NAT64 IPv6 Prefix**— A global IPv6 prefix of length 96 bits (128-32=96) configured on the appliance.
  Note: Currently the NetScaler appliance supports only one prefix to be used commonly with all NAT 64 rules.

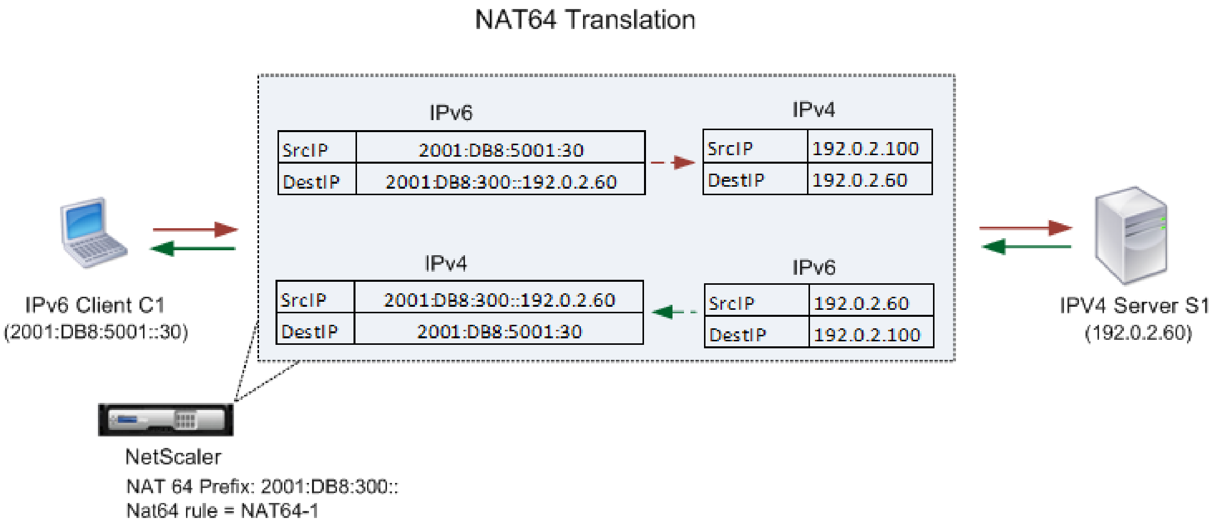
The NetScaler appliance considers an incoming IPv6 packet for NAT64 translation when all of the following conditions are met:

- The incoming IPv6 packet matches the ACL6 rule bound to a NAT64 rule.
- The destination IP address of the IPv6 packet matches the NAT64 IPv6 prefix.

When an IPv6 request packet received by the NetScaler appliance matches an ACL6 defined in a NAT64 rule and the destination IP of the packet matches the NAT64 IPv6 prefix, the NetScaler appliance considers the IPv6 packet for translation.

The appliance translates this IPv6 packet to an IPv4 packet with a source IP address matching one of the IP address bound to the netprofile defined in the NAT64 rule, and a destination IP address consisting of the last 32 bits of the destination IPv6 address of the IPv6 request packet. The NetScaler appliance creates a NAT64 session for this particular flow and forwards the packet to the IPv4 server. Subsequent responses from the IPv4 server and requests from the IPv6 client are translated accordingly by the appliance, on the basis of information in the particular NAT64 session.

Consider an example in which an enterprise hosts site www.example.com on server S1, which has an IPv4 address. To enable communication between IPv6 clients and IPv4 server S1, NetScaler appliance NS1 is deployed with a stateful NAT64 configuration that includes a NAT64 rule and a NAT64 prefix. A mapped IPv6 address of server S1 is formed by concatenating the NAT64 IPv6 prefix [96 bits] and the IPv4 source address [32 bits]. This mapped IPv6 address is then manually configured in the DNS servers. The IPv6 clients get the mapped IPv6 address from the DNS servers to communicate withIPv4 server S1.


NAT64 Translation
IPv6
SrcIP
2001:DB8:5001:30
DestIP
2001:DB8:300::192.0.2.60
IPv4
SrcIP
192.0.2.100
DestIP
192.0.2.60
IPv4
SrcIP
2001:DB8:300::192.0.2.60
DestIP
2001:DB8:5001:30
IPv6
SrcIP
192.0.2.60
DestIP
192.0.2.100
IPv6 Client C1
(2001:DB8:5001::30)
IPV4 Server S1
(192.0.2.60)
NetScaler
NAT 64 Prefix: 2001:DB8:300::
Nat64 rule = NAT64-1

The following table lists the settings used in this example: Stateful NAT64 translation example settings.

Following is the traffic flow in this example:

1. IPv6 client CL1 sends a request packet to Map-Sevr-IPv6 (2001:DB8:300::192.0.2.60) address.
2. The NetScaler appliance receives the request packet. If the request packet matches the ACL6 defined in the NAT64 rule, and the destination IP address of the packet matches the NAT64 IPv6 prefix, the NetScaler considers the IPv6 packet for translation.
3. The appliance creates a translated IPv4 request packet with:
    - Destination IP address field containing the NAT64 prefix stripped from the destination address of the IPv6 request (Sevr_IPv4 = 192.0.2.60)

- Source IP address field containing one of the IPv4 address bound to Netprofile-1(in this case, 192.0.2.100)

4. The NetScaler appliance creates a NAT64 session for this flow and sends the translated IPv4 request to server S1.
5. IPv64 server S1 responds by sending an IPv4 packet to the NetScaler appliance with:
    - Destination IP address field containing 192.0.2.100
    - Source IP address field containing the address ofSevr_IPv4(192.0.2.60)
6. The appliance receives the IPv4 response packet, searches all the session entries, and finds that the IPv6 response packet matches the NAT64 session entry created in step 4. The appliance considers the IPv4 packet for translation.
7. The appliance creates a translated IPv6 response packet with:
    - Destination IP address field=Client_IPv6=2001:DB8:5001::30
    - Source IP address field = Concatenation of NAT64 Prefix (First 96 bits) and Sevr_IPv4 (last 32 bits) =2001:DB8:300::192.0.2.60
8. The appliance sends the translated IPv6 response to client CL1.

Limitations of Statelful NAT64

The following limitations apply to stateful NAT64:

- Translation of IPv4 options is not supported.
- Translation of IPv6 routing headers is not supported.
- Translation of hop-by-hop extension headers of IPv6 packets is not supported.
- Translation of ESP and EH headers of IPv6 packets is not supported.
- Translation of multicast packets is not supported.
- Packets of Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP), and IPSec, are not translated.

Configuring Stateful NAT64

Creating the required entities for stateful NAT64 configuration on the NetScaler appliance involves the following procedures:

1. Add an ACL6 rule with action ALLOW.
2. Add an ipset, which binds multiple IP addresses.
3. Add a netprofile and bind the ipset to it. If you want to bind only one IP address, you need not create an ipset entity. In that case, bind the IP address directly to the netprofile.
4. Add a NAT64 rule, which includes binding the ACl6 rule and the netprofile to the NAT 64 rule.
5. Add a NAT64 IPv6 prefix.

NetScaler command line procedures

To add an ACL6 rule by using the NetScaler command line:

At the command prompt, type:

- add ns acl6 <acl6name> <acl6action> …

To add an IPset and bind multiple IPs to it by using the NetScaler command line:

At the command prompt, type:

- add ipset <name>
- bind ipset <name> <IPaddress …>

To add a netprofile by using the NetScaler command line:

At the command prompt, type:

- add netprofile <name> -srcIP <IPaddress or IPset>

To add a NAT64 rule by using the NetScaler command line:

At the command prompt, type:

- add nat64 <name> <acl6name> -netProfile <string>

To add a NAT64 prefix by using the NetScaler command line:

At the command prompt, type:

```
set ipv6 -natprefix <ipv6_addr                          *>
```

-

Example:

```
 > add acl6 ACL6-1 ALLOW -srcIPv6 2001:DB8:5001::30
 Done

> apply acls6
 Done

> add ip 192.0.2.100 255.255.255.0  - type SNIP
 Done

> add ip 192.0.2.102 255.255.255.0   - type SNIP
 Done

> add ipset IPset-1
```

```
 Done

> bind ipset IPset-1 192.0.2.100 192.0.2.102
IPAddress "192.0.2.100" bound
IPAddress "192.0.2.102" bound
 Done

> add netprofile Netprofile-1 -srcIP IPset-1
 Done

> add nat64 NAT64-1 ACL6-1 -netprofile Netprofile-1
 Done

> set ipv6 -natprefix 2001:DB8:300::/96
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To add a NAT64 rule by using the NetScaler GUI:

Navigate to System > Network > Routes > NAT64, and a new NAT64 rule, or edit an existing rule.

To add a NAT64 prefix by using the NetScaler GUI:

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

RNAT

September 21, 2020

In Reverse Network Address Translation (RNAT), the NetScaler appliance replaces the source IP addresses in the packets generated by the servers with public NAT IP addresses. By default, the appliance uses a SNIP address as the NAT IP address. You can also configure the appliance to use a unique NAT IP address for each subnet. You can also configure RNAT by using Access Control Lists (ACLs). Use Source IP (USIP), Use Subnet IP (USNIP), and Link Load Balancing (LLB) modes affect the operation of RNAT. You can display statistics to monitor RNAT.

Note: The ephemeral port range for RNAT on the NetScaler appliance is 1024-65535.

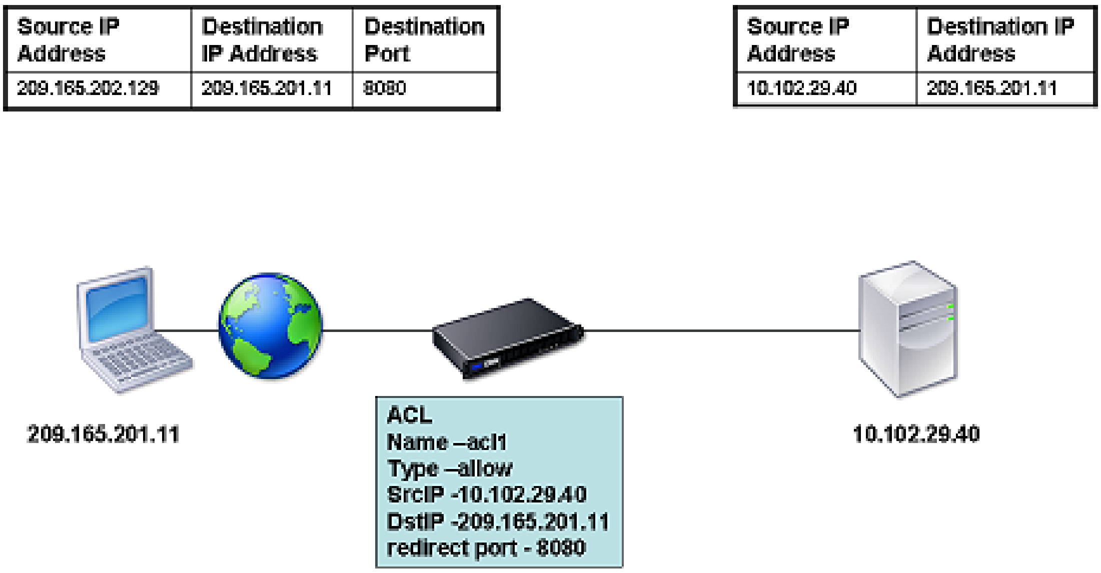
You can use either a network address or an extended ACL as the condition for an RNAT entry:

- **Using a Network address.** When you use a network address, RNAT processing is performed on all of the packets coming from the specified network.
- **Using Extended ACLs.** When you use ACLs, RNAT processing is performed on all packets that match the ACLs. To configure the NetScaler appliance to use a unique IP address for traffic that matches an ACL, you must perform the following three tasks:

  1. Configure the ACL.
  2. Configure RNAT to change the source IP address and Destination Port.
  3. Apply the ACL.

  The following diagram illustrates RNAT configured with an ACL.

  Figure 1. RNAT with an ACL



Source IP Address	Destination IP Address	Destination Port
209.165.202.129	209.165.201.11	8080

Source IP Address	Destination IP Address
10.102.29.40	209.165.201.11

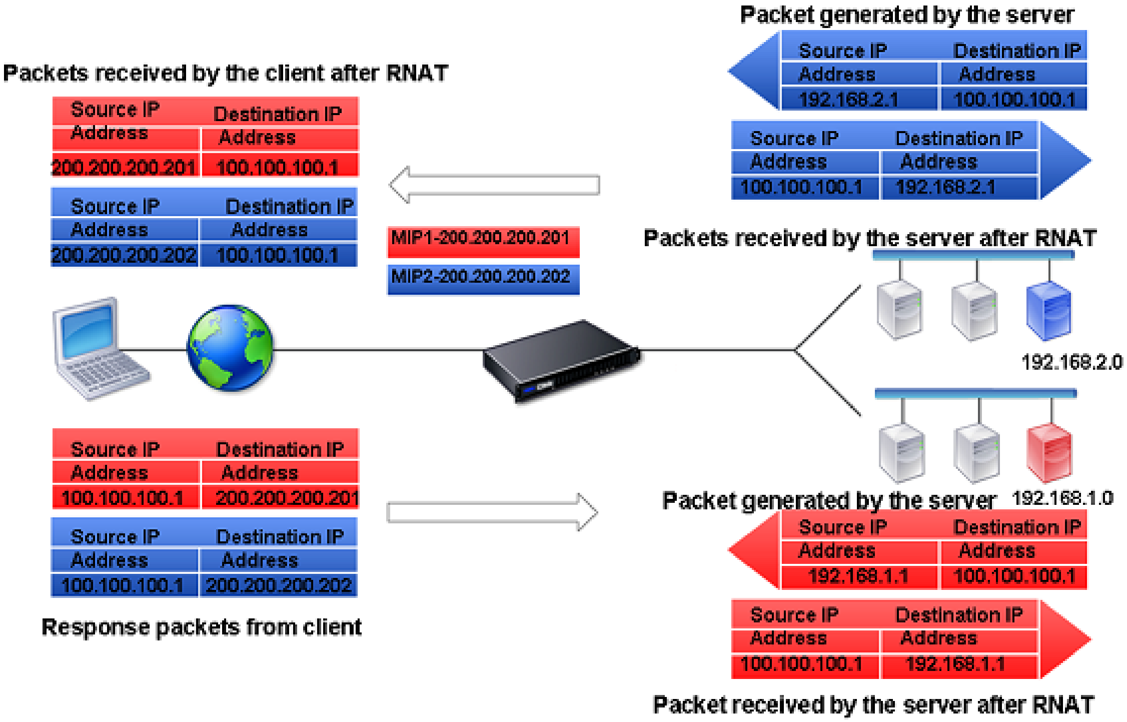
ACL
Name –acl1
Type –allow
SrcIP -10.102.29.40
DstIP -209.165.201.11
redirect port - 8080

You have the following basic choices for the type of NAT IP address:

- **Using a SNIP as the NAT IP Address.** When using a SNIP as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the a SNIP. Therefore, the SNIP address must be a public IP address. If Use Subnet IP (USNIP) mode is enabled, the NetScaler can use a subnet IP address (SNIP) as the NAT IP address.
- **Using a Unique IP Address as the NAT IP Address.** When using a unique IP address as the NAT IP address, the NetScaler appliance replaces the source IP addresses of server-generated packets with the unique IP address specified. The unique IP address must be a public NetScaler-owned IP address. If multiple NAT IP addresses are configured for a subnet, NAT IP selection uses the round robin algorithm.

This configuration is illustrated in the following diagram.

Figure 2. Using a Unique IP Address as the NAT IP Address


Packet generated by the server
Source IP Address
Destination IP Address
192.168.2.1
100.100.100.1
Source IP Address
Destination IP Address
100.100.100.1
192.168.2.1
Packets received by the server after RNAT
Packets received by the client after RNAT
Source IP Address
Destination IP Address
200.200.200.201
100.100.100.1
Source IP Address
Destination IP Address
200.200.200.202
100.100.100.1
MIP1-200.200.200.201
MIP2-200.200.200.202
192.168.2.0
192.168.1.0
Source IP Address
Destination IP Address
100.100.100.1
200.200.200.201
Source IP Address
Destination IP Address
100.100.100.1
200.200.200.202
Response packets from client
Packet generated by the server
Source IP Address
Destination IP Address
192.168.1.1
100.100.100.1
Source IP Address
Destination IP Address
100.100.100.1
192.168.1.1
Packet received by the server after RNAT

Before you begin

Before configuring a RNAT rule, consider the following points:

- When RNAT and Use Source IP (USIP) are both configured on the NetScaler appliance, RNAT takes precedence. In other words, the source IP address of the packets, which matches a RNAT rule, is replaced according to the setting in the RNAT rule.
- In a topology where the NetScaler appliance performs both Link Load Balancing (LLB) and RNAT for traffic originating from the server, the appliance selects the source IP address based on the router. The LLB configuration determines selection of the router. For more information about LLB, see Link Load Balancing.

Configure RNAT

The following instructions provide separate command-line procedures for creating RNAT entries that use different conditions and different types of NAT IP addresses. In the NetScaler GUI, all of the variations can be configured in the same dialog box, so there is only one procedure for NetScaler GUI users.

NetScaler command line procedures

To create an RNAT entry by using the NetScaler command line:

At the command prompt, type one the following commands to create, respectively, an RNAT entry that uses a network address as the condition and a SNIP as the NAT IP address, an RNAT entry that uses a network address as the condition and a unique IP address as the NAT IP address, an RNAT entry that uses an ACL as the condition and a SNIP as the NAT IP address, or an RNAT entry that uses an ACL as a condition and a unique IP address as the NAT IP address:

- **set rnat** <IPAddress> <netmask>
- **set rnat IPAddress** <netMask> -**natip** <NATIPAddress>
- **set rnat** <aclname> [-**redirectPort** <port>]
- **set rnat** <aclname> [-**redirectPort** <port>] -**natIP** <NATIPAddress>

Use the following command to verify the configuration:

- **show rnat**

Examples:

```
A network address as the condition and a  SNIP as the NAT IP address:

> set rnat 192.168.1.0 255.255.255.0
 Done

A network address as the condition and a unique IP address as the NAT IP address:

> set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50
 Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

> set rnat 192.168.1.0 255.255.255.0  -natIP 10.102.29.[50-110]
 Done

An ACL as the condition and a  SNIP as the NAT IP address:

> set rnat acl1
 Done

An ACL as a condition and a unique IP address as the NAT IP address:

> set rnat acl1  -natIP 209.165.202.129
```

```
Done

If instead of a single NAT IP address you specify a range, RNAT entries are created with all the NetScaler-owned IP addresses, except the NSIP, that fall within the range specified:

> set rnat acl1  -natIP 10.102.29.[50-70]
 Done

<!--NeedCopy-->
```

NetScaler GUI procedures

To create an RNAT entry by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. In the **Action list**, select **Configure RNAT** to create or modify an RNAT entry.

Monitor RNAT

You can display RNAT statistics to troubleshoot issues related to IP address translation.

The following table describes the statistics associated with RNAT and RNAT IP.

Statistic	Description
Bytes received	Bytes received during RNAT sessions
Bytes sent	Bytes sent during RNAT sessions
Packets received	Packets received during RNAT sessions
Packets sent	Packets sent during RNAT sessions
Syn sent	Requests for connections sent during RNAT sessions
Current sessions	Currently active RNAT sessions

To view RNAT statistics by using the NetScaler command line:

At the command prompt, type:

- **stat rnat**

Example:

```
> stat rnat

RNAT summary
                                Rate (/s)          Total
Bytes Received                    0                  0
Bytes Sent                        0                  0
Packets Received                  0                  0
Packets Sent                      0                  0
Syn Sent                          0                  0
Current RNAT sessions            --                  0
 Done
>
<!--NeedCopy-->
```

To monitor RNAT by using the NetScaler GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT** tab, and then click **Statistics**.

Configure RNAT6

Reverse Network Address Translation (RNAT) rules for IPv6 packets are called RNAT6s. When an IPv6 packet generated by a server matches the conditions specified in the RNAT6 rule, the appliance replaces the source IPv6 address of the IPv6 packet with a configured NAT IPv6 address before forwarding it to the destination. The NAT IPv6 address is one of the NetScaler owned SNIP6 or VIP6 addresses.

When configuring an RNAT6 rule, you can specify either an IPv6 prefix or an ACL6 as the condition:

- **Using a IPv6 network address**. When you use an IPv6 prefix, the appliance performs RNAT processing on those IPv6 packets whose IPv6 address matches the prefix.
- **Using ACL6s**. When you use an ACL6, the appliance performs RNAT processing on those IPv6 packets that match the conditions specified in the ACL6.

You have one of the following options to set the NAT IP address:

- Specify a set of NetScaler owned SNIP6 and VIP6 addresses for an RNAT6 rule. The NetScaler appliance uses any one of the IPv6 addresses from this set as a NAT IP address for each session. The selection is based on the round robin algorithm and is done for each session.
- Do not specify any NetScaler owned SNIP6 or VIP6 address for an RNAT6 rule. The NetScaler appliance uses any one of the NetScaler owned SNIP6 or VIP6 addresses as a NAT IP address. The selection is based on the next hop network to which an IPv6 packet that matches the RNAT rule is destined.

NetScaler command line procedures

To create an RNAT6 rule by using the NetScaler command line:

At the command prompt, to create the rule and verify the configuration, type:

- **add rnat6** <name> (<network> | (<acl6name> [-**redirectPort** <port>]))
- **bind rnat6** <name> <natIP6>@ …
- **show rnat6**

To modify or remove an RNAT6 rule by using the NetScaler command line:

- To modify an RNAT6 rule whose condition is an ACL6, type the **set rnat6** <name> command, followed by a new value for the **redirectPort** parameter.
- To remove an RNAT6 rule, type the **clear rnat6** <name> command.
- **show rnat6**

NetScaler GUI procedures

To configure an RNAT6 rule by using the NetScaler GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT6** tab, and add a new RNAT6 rule, or edit an existing rule.

Monitor RNAT6

You can display statistics related to the RNAT6 feature to monitor the performance or to troubleshoot problems related to RNAT6 feature. You can display a summary of statistics of the RNAT6 rules or of a particular RNAT6 rule. The statistical counters reflect events since the NetScaler appliance was last restarted. All these counters are reset to 0 when the NetScaler appliance is restarted.

The following lists some of the statistics counters associated with the RNAT6 feature:

- **Bytes received** - Total bytes received during RNAT6 sessions.
- **Bytes sent** - Total number of bytes sent during RNAT6 sessions.
- **Packets received** - Total number of packets received during RNAT6 sessions.
- **Packets sent** - Total number of packets sent during RNAT6 sessions.
- **Syn sent** - Total number of requests for connections sent during RNAT6 sessions
- **Current sessions** - Currently active RNAT6 sessions

To display a summarized statistics of all RNAT6 rules by using the NetScaler command line:

At the command prompt, type:

- **stat rnat6**

To display statistics for a specified RNAT6 rule by using the NetScaler command line:

At the command prompt, type:

- **stat rnat6** [<rnat6 rule name>]

To display RNAT6 statistics by using the NetScaler GUI:

Navigate to **System** > **Network** > **NATs**, click the **RNAT6** tab, and then click **Statistics**.

```
> stat rnat6

RNAT6 summary

                           Rate (/s)                Total

Bytes Received                178                20644

Bytes Sent                    178                20644

Packets Received                5                  401

Packets Sent                    5                  401

Syn Sent                        0                    2

Current RNAT6 sessions       --                      1

Done

<!--NeedCopy-->
```

Log Start Time and Connection Closure Reasons in RNAT Log Entries

For diagnosing or troubleshooting problems related to RNAT, the NetScaler appliance logs RNAT sessions whenever they are closed.

A log message for an RNAT session consists of the following information:

- NetScaler owned IP address (NSIP address or SNIP address) from which the log message is sourced
- Time stamp of log creation
- Protocol of the RNAT session
- Source IP address
- RNAT IP address
- Destination IP address
- Start time of the RNAT session
- Closing time of the RNAT session
- Total bytes sent by the NetScaler appliance for this RNAT session
- Total bytes received by the NetScaler appliance for this RNAT session

- Reason for closure of the RNAT session. The NetScaler appliance logs closure reason for TCP RNAT sessions that do not use the TCP proxy (TCP proxy disabled) of the appliance. The following are the type of closure reasons that are logged for TCP RNAT sessions:
  - **TCP FIN**. The RNAT session was closed because of a TCP FIN sent by either the source or destination device.
  - **TCP RST**. The RNAT session was closed because of a TCP Reset that was sent by either the source or destination device.
  - **TIMEOUT**. The RNAT session timed out.

The following table shows some sample log entries for RNAT sessions.

Type of Entry	Sample Log Entry
Sample log entry for UDP RNAT session	Dec 1 15:28:12 10.102.53.114 12/01/2015:15:28:12 GMT 0-PPE-0 : default UDP NAT_OTHERCONN_DELINK 154 0 : Source 1.2.2.5:23431 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:4045 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:26:58 GMT - Delink Time 12/01/2015:15:28:12 GMT - Total_bytes_send 2511 - Total_bytes_recv 3725
Sample log entry for TCP RNAT session. The log entry shows that the session closed because of TCP Reset	Dec 1 15:29:59 10.102.53.114 12/01/2015:15:27:59 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 152 0 : Source 1.2.2.5:33826 - Destination 192.168.123.122:22 - NatIP 192.168.123.1:2384 - Destination 192.168.123.122:22 - Start Time 12/01/2015:15:27:40 GMT - Delink Time 12/01/2015:15:27:59 GMT - Total_bytes_send 2147 - Total_bytes_recv 3257 - Closure Reason TCP RST

Type of Entry	Sample Log Entry
Sample log entry for TCP RNAT session. The log entry shows that the session timed out	Dec 1 15:30:12 10.102.53.114 12/01/2015:15:30:12 GMT 0-PPE-0 : default TCP NAT_OTHERCONN_DELINK 155 0 : Source 1.2.2.5:64976 - Destination 192.168.123.115:22 - NatIP 192.168.123.1:19636 - Destination 192.168.123.115:22 - Start Time 12/01/2015:15:27:25 GMT - Delink Time 12/01/2015:15:30:12 GMT - Total_bytes_send 0 - Total_bytes_recv 0 - Closure Reason TIMEOUT

Stateful Connection Failover for RNAT

Connection failover helps prevent disruption of access to applications deployed in a distributed environment. The NetScaler appliance now supports stateful connection failover for connections related to RNAT rules in a NetScaler High Availability (HA) setup. In an HA setup, connection failover (or connection mirroring) refers to the process of keeping an established TCP or UDP connection active when a failover occurs.

The primary appliance sends messages to the secondary appliance to synchronize current information about the RNAT connections. The secondary appliance uses this connection information only in the event of a failover. When a failover occurs, the new primary NetScaler appliance has information about the connections established before the failover and hence continues to serve those connections even after the failover. From the client's perspective this failover is transparent. During the transition period, the client and server may experience a brief disruption and retransmissions.

Connection failover can be enabled per RNAT rule. For enabling connection failover on an RNAT rule, you enable the connFailover (Connection Failover) parameter of that specific RNAT rule by using either NetScaler command line or NetScaler GUI.

To enable connection failover for a RNAT rule by using the NetScaler command line:

At the command prompt, type:

- **set rnat** <name> -**connfailover** (**ENABLED** | **DISABLED**)
- **show rnat**

To enable connection failover for a RNAT rule by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. Select **Connection Failover** while adding a new RNAT rule, or while editing an existing rule.

Removing RNAT Sessions

You can remove any unwanted or inefficient RNAT sessions from the NetScaler appliance. The appliance immediately releases resources (such as the port of the NAT IP address, and memory) allocated for these sessions, making the resources available for new sessions. The appliance also drops all the subsequent packets related to these removed sessions. You can remove all or selected RNAT sessions from the NetScaler appliance.

To clear all RNAT sessions by using the NetScaler command line:

At the command prompt, type:

- **flush rnatsession**

To clear selective RNAT sessions by using the NetScaler command line:

At the command prompt, type:

- **flush rnatsession** ((-**network** <ip_addr> -**netmask** <netmask>) | -**natIP** <ip_addr> | -**aclname** <string>)

To clear all or selective RNAT sessions by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **NATs**, and then click the **RNAT** tab.
2. In the **Actions** Menu, click **Flush RNAT Sessions** to remove all or selective RNAT sessions (for example, removing RNAT sessions with a specific RNAT IP, or belonging to a specific Network or ACL based RNAT rule).

Sample Configurations:

```
Clear all RNAT sessions existing on a NetScaler appliance

> flush rnatsession

Done

Clear all RNAT sessions belonging to network based RNAT rules that has
    203.0.113.0/24 network as the matching condition.

> flush rnatsession -network 203.0.113.0 -netmask 255.255.255.0

Done

Clear all RNAT sessions with RNAT IP 192.0.2.90.

> flush rnatsession -natIP 192.0.2.90

Done

```

```
Clear all RNAT sessions belonging to ACL based RNAT rules that has ACL-RNAT-1 as the matching condition.

> flush rnatsession -aclname ACL-RNAT-1

Done
<!--NeedCopy-->
```

Configuring Prefix-Based IPv6-IPv4 Translation

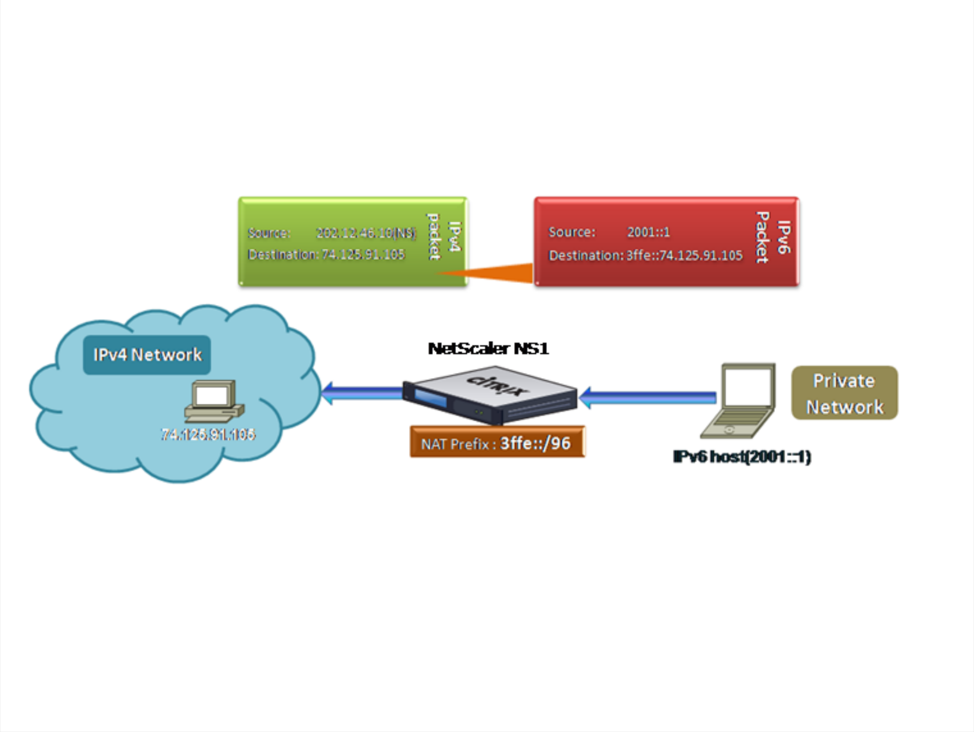
September 21, 2020

Prefix-based translation is a process of translating packets sent from private IPv6 servers into IPv4 packets, using an IPv6 prefix configured in the NetScaler appliance. This prefix has a length of 96 bits (128-32=96). The IPv6 servers embed the destination IP address of the IPv4 servers or hosts in the last 32 bits of the destination IP address field of the IPv6 packets. The first 96 bits of the destination IP address field are set as the IPv6 NAT prefix.

The NetScaler appliance compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix. If there is a match, the NetScaler appliance generates an IPv4 packet and sets the destination IP address as the last 32 bits of the destination IP address of the matched IPv6 packet. IPv6 packets addressed to this prefix have to be routed to the NetScaler so that the IPv6-IPv4 translation is done by the NetScaler.

In the following diagram, 3ffe::/96 is configured as the IPv6 NAT prefix on NetScaler NS1. The IPv6 host sends an IPv6 packet with destination IP address 3ffe::74.125.91.105. NS1 compares the first 96 bits of the destination IP address of all the incoming IPv6 packets to the configured prefix, and they match. NS1 then generates an IPv4 packet and sets the destination IP address as 74.125.91.105.

Figure 1. IPv6-IPv4 Prefix-Based Translation


Source: 202.12.46.10(NS)
Destination: 74.125.91.105
IPv4 packet
Source: 2001::1
Destination: 3ffe::74.125.91.105
IPv6 Packet
IPv4 Network
74.125.91.105
NetScaler NS1
NAT Prefix : 3ffe::/96
Private Network
IPv6 host(2001::1)

To configure prefix-based IPv6-IPv4 translation by using the NetScaler command line:

At the command prompt, type:

```
set ipv6 [-natprefix <ipv6_addr                    *>]
```

-
- show ipv6

Example:

```
> set ipv6 -natprefix 3ffe::/96
 Done
<!--NeedCopy-->
```

To configure prefix-based IPv6-IPv4 translation by using the NetScaler GUI:

Navigate to System > Network, in the Settings group, click Configure INAT Parameters, and set the Prefix parameter.

IP Prefix NAT

September 21, 2020

The NetScaler appliance supports translating a part of the source IP address instead of the complete address of packets received on the appliance. IP prefix NAT includes changing one or more octets or bits of the source IP address.

The NetScaler appliance supports IP prefix NAT for traffic related to virtual servers and services for which the NetScaler does not maintain any session information. For example, virtual servers and services of type ANY, UDP, and DNS.

Use Case: Zonification of Clients for a Deployment of a NetScaler appliance and an Optimization Device

IP prefix NAT is very useful in a deployment that includes a NetScaler appliance and an optimization device (for example, Citrix ByteMobile). This type of deployment has different geographically located client networks, which share the same network address. The NetScaler appliance must send the traffic received from each of the client networks to the optimization device before forwarding to the destination.

The device sends the optimized traffic back to the NetScaler appliance. Because the optimization requirement is different for traffic from each client network, the optimization device must recognize the client network of each packet that it receives. The solution is to segregate traffic from each client network into a different zone by using VLANs. IP prefix NAT with a different setting is configured for each zone. The NetScaler appliance translates the last octet of the source IP address of every packet, and the translated octet value is different for each zone.

Consider an example of two zones, Z1 and Z2, sharing network address 192.0.2.0/24. On the NetScaler appliance, IP prefix NAT entities named natrule-1 and natrule-2 are configured for these two zones. Before the appliance forwards a packet from Z1, natrule-1 translates the last octet of the packet's source IP address to 100. Similarly, for packets from Z2, natrule-2 translates the last octet of the source IP address to 200. For two clients, CL1-Z1 in zone Z1 and CL1-Z2 in zone Z2, each with IP address 192.0.2.30, the NetScaler appliance translates the source IP address of CL1-Z1's packets to 100.0.2.30 and of CL1-Z2's packets to 200.0.2.30. The optimization device to which the NetScaler appliance sends the translated packets is configured to use a packet's source IP address to recognize the zone, so it applies the appropriate optimization configured for the zone from which the packet originated.

Configuration Steps

Configuring IP prefix NAT consists of the following steps:

- **Create a net profile and set the NAT Rule parameter of a net profile**. A NAT rule specifies two IP addresses and a net mask. The first IP address (specified by IP Address parameter) is the source IP address that is to be translated with the second one (specified by IP Rewrite parameter). The net mask specifies the part of the source IP address that is to be translated with the same part of the second IP address.

- **Bind the net profile to load balancing virtual servers or services**. A net profile with NAT rule setting can be bound to a virtual server or service of only type ANY, UDP, or DNS. After binding a net profile to a virtual server or service, the NetScaler appliance matches the source IP address of the incoming packets related to the virtual server or service with the NAT rule setting. The NetScaler then performs IP prefix NAT for packets that match the NAT rule.

To configure IP prefix NAT translation by using the command line:

At the command prompt, type:

- **bind netProfile** <name> (-**natRule** <ip_addr> <netmask> <rewriteIp>)
- **show netprofile** <name>

To configure IP prefix NAT by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **Net Profiles**.
2. Set the following parameters under NAT Rules while adding or modifying NetProfiles.
    - IP Address
    - Netmask
    - Rewrite IP

Sample configuration

In the following sample configuration, net profile PARTIAL-NAT-1 has IP prefix NAT settings and is bound to load balancing virtual server LBVS-1, which is of type ANY. For packets received on LBVS-1 from 192.0.0.0/8, the NetScaler appliance translates the last octet of the packet's source IP address to 100. For example, a packet with source IP address 192.0.2.30 received on LBVS-1, the NetScaler appliance translates the source IP address to 100.0.2.30 before sending it one of the bound servers.

```
> add netprofile PARTIAL-NAT-1
Done

> bind netprofile PARTIAL-NAT-1 -natrule 192.0.0.0 255.0.0.0 100.0.0.0
Done

> add lb vserver LBVS-1 ANY 203.0.113. 61 * -netprofile PARTIAL-NAT-1
Done
<!--NeedCopy-->
```

Static ARP

September 21, 2020

You can add static ARP entries to and remove static ARP entries from the ARP table. After adding an entry, you should verify the configuration. If the IP address, port, or MAC address changes after you create a static ARP entry, you must remove or manually adjust the static entry. Therefore, creating static ARP entries is not recommended unless necessary.

To add a static ARP entry by using the NetScaler command line:

At the command prompt, type:

- **add arp -IPAddress** <ip_addr> -**mac**<mac_addr> -**ifnum** <interface_name>
- **show arp** <IPAddress>

Example:

```
> add arp -ip 10.102.29.6 -mac 00:24:e8:73:ca:ec -ifnum 1/1
 Done
<!--NeedCopy-->
```

To remove a static ARP entry by using the NetScaler command line:

At the command prompt, type the **rm arp** command and the IP address.

To add a static ARP entry by using the NetScaler GUI:

Navigate to **System** > **Network** > **ARP Table**, and add a static ARP entry.

Specify a VLAN in a Static ARP Entry

In a static ARP entry, you can specify the VLAN through which the destination device is accessible. This feature is useful when the interface specified in the static ARP entry is part of multiple tagged VLANs and the destination is accessible through one of the VLANs. The NetScaler appliance includes the specified VLAN ID in the outgoing packets matching the static ARP entry. If you don't specify a VLAN ID in an ARP entry, and the specified interface is part of multiple tagged VLANs, the appliance assigns the interface's native VLAN to the ARP entry.

For example, say NetScaler interface 1/2 is part of native VLAN 2 and of tagged VLANs 3 and 4, and you add a static ARP entry for network device A, which is part of VLAN 3 and is accessible through interface 1/2. You must specify VLAN 3 in the ARP entry for network device A. The NetScaler appliance then includes tagged VLAN 3 in all the packets destined to network device A, and sends them from interface 1/2.

If you don't specify a VLAN ID, the NetScaler appliance assigns native VLAN 2 for the ARP entry. Packets destined to device A are dropped in the network path, because they do not specify tagged VLAN 3, which is the VLAN for device A.

To specify a VLAN in a static ARP entry by using the NetScaler command line:

At the command prompt, type:

- **add arp -IPAddress** <ip_addr> **-mac**<mac_addr> **-ifnum** <interface_name> [**-vlan** <positive_integer>]
- **show arp** <IPAddress>

Example:

```
> add arp -ip 198.51.100.91 -mac 36:db:4b:f6:12:15 -ifnum 1/2 -vlan 3
Done
<!--NeedCopy-->
```

Set the Timeout for Dynamic ARP Entries

September 21, 2020

You can globally set an aging time (time-out value) for dynamically learned ARP entries. The new value applies only to ARP entries that are dynamically learned after the new value is set. Previously existing ARP entries expire after the previously configured aging time. You can specify an ARP time-out value of from 1 through 1200 seconds.

To set the time-out for dynamic ARP entries by using the NetScaler command line:

At the command prompt, type:

- **set arpparam** -**timeout** <positive_integer>]
- **show arpparam**

Example:

```
> set arpparam -timeout 500
 Done
<!--NeedCopy-->
```

To set the time-out for dynamic ARP entries to its default value by using the NetScaler command line:

At the command prompt, type:

- **unset arpparam**
- **show arpparam**

Example:

```
> unset arpparam
 Done
<!--NeedCopy-->
```

To set the time-out for dynamic ARP entries by using the NetScaler GUI:

Navigate to **System** > **Network**, in the **Settings** group, click **Configure ARP Global Parameters**, and set the **ARP Table Entry Timeout** parameter.

Neighbor Discovery

September 21, 2020

Neighbor discovery (ND) is one of the most important protocols of IPv6. It is a message-based protocol that combines the functionality of the Address Resolution Protocol (ARP), Internet Control Message Protocol (ICMP), and Router Discovery. ND allows nodes to advertise their link layer addresses and obtain the MAC addresses or link layer addresses of the neighboring nodes. This process is performed by the Neighbor Discovery protocol (ND6).

Neighbor discovery can perform the following functions:

- **Router Discovery**: Enables a host to discover the local routers on an attached link and automatically configure a default router.
- **Prefix Discovery**: Enables the host to discover the network prefixes for local destinations.

  Note: The NetScaler appliance does not support Prefix Discovery.
- **Parameter Discovery**: Enables a host to discover additional operating parameters, such as MTU and the default hop limit for outbound traffic.
- **Address Autoconfiguration**: Enables hosts to automatically configure IP addresses for interfaces both with and without stateful address configuration services such as DHCPv6. The NetScaler does not support Address Autoconfiguration for Global IPv6 addresses.
- **Address Resolution**: Equivalent to ARP in IPv4, enables a node to resolve a neighboring node's IPv6 address to its link-layer address.
- **Neighbor Unreachability Detection**: Enables a node to determine the reachability state of a neighbor.
- **Duplicate Address Detection**: Enables a node to determine whether an NSIP address is already in use by a neighboring node.
- **Redirect**: Equivalent to the IPv4 ICMP Redirect message, enables a router to redirect the host to a better first-hop IPv6 address to reach a destination.

Note: The NetScaler appliance does not support IPv6 Redirect.

Configuration Steps

Configuring neighbor discovery consists of the following tasks:

- Adding IPv6 neighbors
- (Optional) Removing IPv6 neighbors

NetScaler command line procedures

To add an IPv6 neighbor by using the NetScaler command line:

At the command prompt, type:

- **add nd6** <neighbor> <mac> <ifnum> [-**vlan** <integer>]
- **sh nd6**

Example:

```
> add nd6 2001::1 00:04:23:be:3c:06 1/1  - vlan 1
 Done

> show nd6
   Neighbor                 MAC-Address(Vlan, Interface)       State
             TIME
   --------                 ----------------------------       -----
           --------
1) ::1                      00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
       PERMANENT
2) fe80::2d0:68ff:fe0b:58da 00:d0:68:0b:58:da(   1,  LO/1)   REACHABLE
       PERMANENT
3) 2001::1                  00:04:23:be:3c:06(   1,   1/1)   REACHABLE
         STATIC
 Done
<!--NeedCopy-->
```

To remove a neighbor discovery entry by using the NetScaler command line:

At the command prompt, type:

- **rm nd6** <Neighbor> -**vlan** <VLANID>

Example:

```
 rm nd6 3ffe:100:100::1 -vlan 1
<!--NeedCopy-->
```

To remove all neighbor discovery entries by using the NetScaler command line:

At the command prompt, type:

- **clear nd6**

NetScaler GUI procedures

To add an IPv6 neighbor by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, and add a new IPv6 neighbor.

To remove a neighbor discovery entry by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, delete the IPv6 neighbor.

To remove all neighbor discovery entries by using the NetScaler GUI:

Navigate to **System** > **Network** > **IPv6 Neighbors**, and click **Clear**.

IP Tunnels

September 21, 2020

An IP Tunnel is a communication channel, that can be created by using encapsulation technologies, between two networks that do not have a routing path. Every IP packet that is shared between the two networks is encapsulated within another packet and then sent via the tunnel.

The NetScaler appliance implements IP Tunneling in the following ways:

- **NetScaler as an Encapsulator (Load Balancing with DSR Mode)**: Consider an organization that has multiple data centers across different countries, where the NetScaler maybe at one location and the back-end servers are located in a different country. In essence, the NetScaler and the back-end servers are on different networks and are connected via a router.

  When you configure Direct Server Return (DSR) on this NetScaler, the packet sent from the source subnet is encapsulated by the NetScaler and sent via a router and tunnel to the appropriate back-end server. The back-end server decapsulates the packet and responds directly to the client, without allowing the packet to pass via the NetScaler.

- **NetScaler as a Decapsulator:** Consider an organization having multiple data centers each having NetScalers and back-end servers. When a packet is sent from data center A to data center B it is usually sent via an intermediary, say a router or another NetScaler. The NetScaler processes the packet and then forwards the packet to the back-end server. However, if an encapsulated packet is sent, the NetScaler must be able to decapsulate the packet before sending it to the back-end servers. To enable the NetScaler to function as a decapsulator, a tunnel is added between the router and the NetScaler. When the encapsulated packet, with additional header information, reaches the NetScaler, the data packet is decapsulated i.e. the additional header information is removed, and the packet is then forwarded to the appropriate back-end servers.

  The NetScaler can also be used as a decapsulator for the Load Balancing feature, specifically in scenarios when the number of connections on a vserver exceeds a threshold value and all the new connections are then diverted to a back-up vserver.

Configure IP Tunnels

Configuring IP tunnels on a NetScaler appliance consists of creating IP tunnel entities. An IP tunnel entity specifies the local and remote tunnel end-point IP addresses and the protocol to be used for the IP tunnel.

Note: While configuring an IP tunnel in a cluster setup, the local IP address must be a striped SNIP address.

NetScaler command line procedures

To create an IP tunnel by using the NetScaler command line:

At the command prompt type:

- **add iptunnel** <name> <remote> <remoteSubnetMask> <local> -**type** -**protocol** (**ipoverip** | **GRE**)
- **show iptunnel**

To remove an IP tunnel by using the NetScaler command line:

To remove an IP tunnel, type the **rm iptunnel** command and the name of the tunnel.

To create an IPv6 tunnel by using the NetScaler command line:

At the command prompt type:

- **add ip6tunnel** <name> <remoteIp> <local>
- **show ip6tunnel**

To remove an IPv6 tunnel by using the NetScaler command line:

To remove an IPv6 tunnel, type the **rm ip6tunnel** command and the name of the tunnel.

NetScaler GUI procedures

To create an IP Tunnel by using the NetScaler GUI:

Navigate to **System** > **Network** > **IP Tunnels**, add a new IP tunnel.

To create an IPv6 Tunnel by using the NetScaler GUI:

Navigate to **System** > **Network** > **IP Tunnels** > **IPv6 Tunnels**, and add a new IPv6 tunnel.

Customizing IP Tunnels Globally

By globally specifying the source IP address, you can assign a common source IP address across all tunnels. Also, because fragmentation is CPU-intensive, you can globally specify that the NetScaler appliance drop any packet that requires fragmentation. Alternatively, if you would like to fragment all packets as long as a CPU threshold value is not reached, you can globally specify the CPU threshold value.

NetScaler command line procedures

To globally customize IP tunnels by using the NetScaler command line:

At the command prompt, type:

- **set ipTunnelParam** -**srcIP** <sourceIPAddress> -**srcIPRoundRobin** (**YES** | **NO**)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ipTunnelParam**

Example:

```
> set iptunnelparam - srcIP 12.12.12.22 -dropFrag Yes -dropFragCpuThreshold 50
 Done

> set iptunnelparam -srcIPRoundRobin YES -dropFrag Yes -dropFragCpuThreshold 50
 Done
<!--NeedCopy-->
```

To globally customize IPv6 tunnels by using the NetScaler command line:

At the command prompt, type:

- **set ip6tunnelparam** -**srcIP** <IPv6Address> -**srcIPRoundRobin** (**YES** | NO)-**dropFrag** [**YES** | **NO**] -**dropFragCpuThreshold** <Positive integer>
- **show ip6tunnelparam**

NetScaler GUI procedures

To globally customize IP tunnels by using the NetScaler GUI:

Navigate to **System** > **Network**, in the Settings group, click **IPv4 Tunnel Global Settings**.

1. Navigate to **System** > **Network**, in the **Settings** group, click **IPv6 Tunnel Global Settings**.
2. In the **Configure IP Tunnel Global Parameters** dialog box, set the parameters.

To globally customize IPv6 tunnels by using the NetScaler GUI:

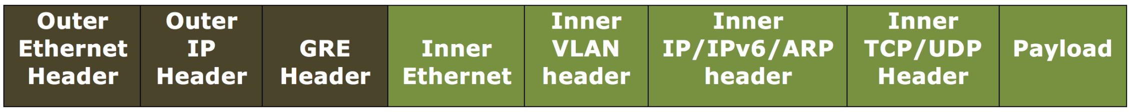
1. Navigate to **System** > **Network**, in the **Settings** group, click **IPv6 Tunnel Global Settings**.
2. In the **Configure IP Tunnel Global Parameters** dialog box, set the parameters.

GRE Payload Options in a GRE IP Tunnel

For a configured GRE IP tunnel, the NetScaler appliance encapsulates the entire Layer 2 packet, including the Ethernet header and the VLAN header (dot1q VLAN tag). IP GRE tunnels between NetScaler appliances and some 3rd party devices might not be stable, because these 3rd party devices are not programmed to process some or the Layer 2 packet headers. To configure a stable IP GRE tunnel between a NetScaler appliance and a 3rd party device, you can use the GRE payload parameter of the GRE IP tunnel command set. GRE payload setting can also be applied to a GRE with IPSec tunnel.

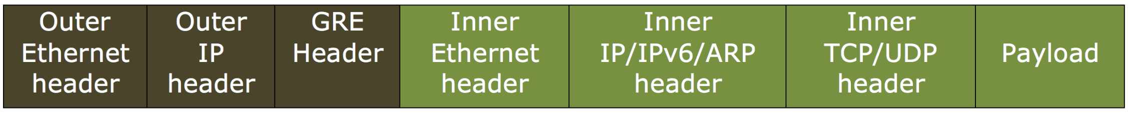
You can set the GRE payload parameter to do one of the following before the packet is sent through the GRE tunnel:

- **Ethernet with DOT1Q**. Carry the Ethernet header as well the VLAN header. This is the default setting. For a tunnel bound to a netbridge, inner Ethernet header and VLAN header contains information from the ARP and bridge table of the NetScaler appliance. For a tunnel set as a next hop to a PBR rule, Inner Ethernet destination MAC address is set to zero and the VLAN header specifies the default VLAN. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:



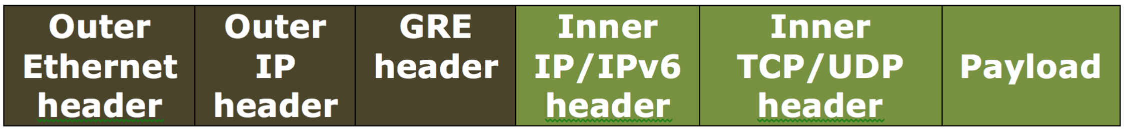
Outer Ethernet Header	Outer IP Header	GRE Header	Inner Ethernet	Inner VLAN header	Inner IP/IPv6/ARP header	Inner TCP/UDP Header	Payload

- **Ethernet**. Carry the Ethernet header but drop the VLAN header. Because the packets do not carry any VLAN information in the tunnel, for a tunnel with this setting and bound to a netbridge, you must bind an appropriate VLAN to the netbridge so that on receiving any packets on the tunnel, the NetScaler can forward these packet to the specified VLAN. If the tunnel is set as a next hop in a PBR rule, the NetScaler routes the packets that are received on the tunnel. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:



Outer Ethernet header	Outer IP header	GRE Header	Inner Ethernet header	Inner IP/IPv6/ARP header	Inner TCP/UDP header	Payload

- **IP**. Drop the Ethernet header as well as the VLAN header. Because tunnels with this setting do not carry Layer 2 headers, these tunnels cannot be bound to a netbridge but can be set as a next hop in a PBR rule. The peer tunnel endpoint device on receiving the packet either consumes or routes it. The encapsulated (GRE) packet sent from the NetScaler tunnel end point has the following format:



To drop Layer 2 headers of packets in a GRE IP tunnel by using the NetScaler command line:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> [-**protocol** <GRE> [-**vlan** <positive_integer>]] [-**grepayload** <grepayload>] [-**ipsecProfileName** <string>]
- **show iptunnel** <tunnelname>

Example:

```
> add iptunnel IPTUNNEL-1 203.0.113.133 255.255.255.0 198.51.100.15 -
    protocol GRE  - grepayload Ethernet -ipsecProfileName IPTUNNEL-IPSEC
    -1
Done
<!--NeedCopy-->
```

IPv6 Traffic through GRE IPV4 Tunnels

The NetScaler appliance supports transferring IPv6 traffic through an IPV4 GRE tunnel. This feature can be used for enabling communication between Isolated IPv6 networks without upgrading the IPv4 infrastructure between them.

For configuring this feature, you associate a PBR6 rule with the configured IPv4 GRE tunnel through which you want the NetScaler to send and receive IPv6 traffic. The source IPv6 address and destination IPv6 address parameters of the PBR6 rule specify the IPv6 networks whose traffic is to traverse the IPv4 GRE tunnel.

Note: IPSec protocol is not supported on GRE IPv4 tunnels that are configured to transfer IPv6 packets.

To create a GRE IPv4 tunnel by using the NetScaler command line:

At the command prompt, type:

- **add ipTunnel** <name> <remote> <remoteSubnetMask> <local> -**protocol GRE**
- **show ipTunnel** <name>

To associate a PBR6 rule with a GRE IPv4 tunnel by using the NetScaler command line:

- **add ns pbr6** <pbrName> **ALLOW** -**srcIPv6** <network-range> -**dstIPv6** <network-range> -**ipTunnel** <tunnelName>
- **show pbr**

Sample configuration

In the following sample configuration, GRE IP tunnel TUNNEL-V6onV4 is created with remote tunnel endpoint IP address 10.10.6.30 and local tunnel endpoint IP address 10.10.5.30. The tunnel is then bound to pbr6 PBR6-V6onV4. The srcIPv6 specifies the IPv6 network connected to the local endpoint and destIPv6 specifies the IPv6 network connected to the remote endpoint. The traffic from these Ipv6 networks are allowed to traverse through the GRE IPv4 tunnel.

```
> add ipTunnel TUNNEL-V6onV4 10.10.6.30 255.255.255.255 10.10.5.30 -protocol GRE
-ipsecProfileName None
Done
> add ns pbr6 PBR6-V6onV4 ALLOW -srcIPv6 = 2001:0db8:1::1-2001:0db8:1::255 -destIPv6 =
1-2001:0db8:4::255 -ipTunnel TUNNEL-V6onV4
<!--NeedCopy-->
```

Send Response Traffic Through an IP-IP Tunnel

You can configure a NetScaler appliance to send response traffic through an IP-IP tunnel instead of routing it back to the source. By default, when the appliance receives a request from another NetScaler or a third-party device through an IP-IP tunnel, it routes the response traffic instead of sending it through the tunnel. You can use policy based routes (PBRs) or enable MAC-Based Forwarding (MBF) to send the response through the tunnel.

In a PBR rule, specify the subnets at both end points whose traffic is to traverse the tunnel. Also set the next hop as the tunnel name. When response traffic matches the PBR rule, the NetScaler appliance sends the traffic through the tunnel.

Alternatively, you can enable MBF to meet this requirement, but the functionality is limited to traffic for which the NetScaler appliance stores session information (for example, traffic related to load balancing or RNAT configurations). The appliance uses the session information to send the response traffic through the tunnel.

NetScaler command line procedures

To create a PBR rule and associate the IP-IP tunnel to it by using the NetScaler command line:

At the command prompt, type:

- **add ns pbr** <pbr_name> **ALLOW** -**srcIP** = <local_subnet_range> -**destIP** = <remote_subnet_range> -**ipTunnel** <tunnel_name>
- **apply ns pbrs**
- **show ns pbr** <pbr_name>

To enable MAC-based forwarding by using the NetScaler command line:

At the command prompt, type:

- **enable ns mode MBF**
- **show ns mode**

NetScaler GUI procedures

To create a PBR rule and associate the IP-IP tunnel to it by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **PBRs**. On the **PBRs** tab, create a **PBR** rule.
2. While creating the PBR, set the **Next Hop Type** to **IP tunnel** and **IP Tunnel Name** to the configured IP-IP tunnel name.

To enable MAC-based forwarding by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features**, click **Configure Modes**.
2. On the **Configure Modes** page, select **MAC-based forwarding**.

Sample configuation

Consider an example of an IPIP tunnel, NS1-NS2-IPIP, which is set up between two NetScaler appliances NS1 and NS2.

By default, for any request that NS2 receives through the tunnel, its routes the response traffic to the source instead of sending it (to NS1) through the tunnel.

You can configure policy based routes (PBRs) or enable MAC-Based Forwarding (MBF) on NS2 for enabling it to send the response through the tunnel.

In the following sample configuration on NS2, NS1-NS2-IPIP is an IPIP tunnel and NS1-NS2-IPIP-PBR is a PBR rule. For requests (with inner source IP address in the range 10.102.147.0-10.102.147.255 and inner destination IP address in the range 10.102.147.0-10.102.147.255) received by NS2 through the tunnel, NS2 sends the corresponding response through the tunnel (to NS1) instead of routing it to the source. The functionality is limited to the traffic that matches the PBR rule.

```
> add iptunnel NS1-NS2-IPIP 192.0.2.99 255.255.255.255 203.0.113.99 -protocol IPIP

Done
> add pbr NS1-NS2-IPIP-PBR -srcIP 10.102.147.0-10.102.147.255  -destIP 10.20.1.0-10.20.1.255  -ipTunnel NS1-NS2-IPIP

Done
> apply pbrs
```

```

Done
```

Alternatively, MBF can be enabled on NS2. The functionality is limited to traffic for which NS2 stores session information (for example, traffic related to load balancing or RNAT configurations).

```
> enable ns mode MBF

Done
```

Interfaces

September 21, 2020

Before you begin configuring interfaces, decide whether your configuration can use MAC-based forwarding mode, and either enable or disable this system setting accordingly. The number of interfaces in your configuration is different for the different models of the Citrix NetScaler appliance. In addition to configuring individual interfaces, you can logically group interfaces, using VLANs to restrict data flow within a set of interfaces, and you can aggregate links into channels. In a high availability setup, you can configure a virtual MAC address if necessary. If you use L2 mode, you might want to modify the aging of the bridge table.

When your configuration is complete, decide whether you should enable the system setting for path MTU discovery. NetScaler appliances can be deployed in active-active mode using VRRP. An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. You can use the Network Visualizer tool to view the network configuration of a NetScaler deployment and configure interfaces, channels, VLANs, and bridge groups.

Configuring MAC-Based Forwarding

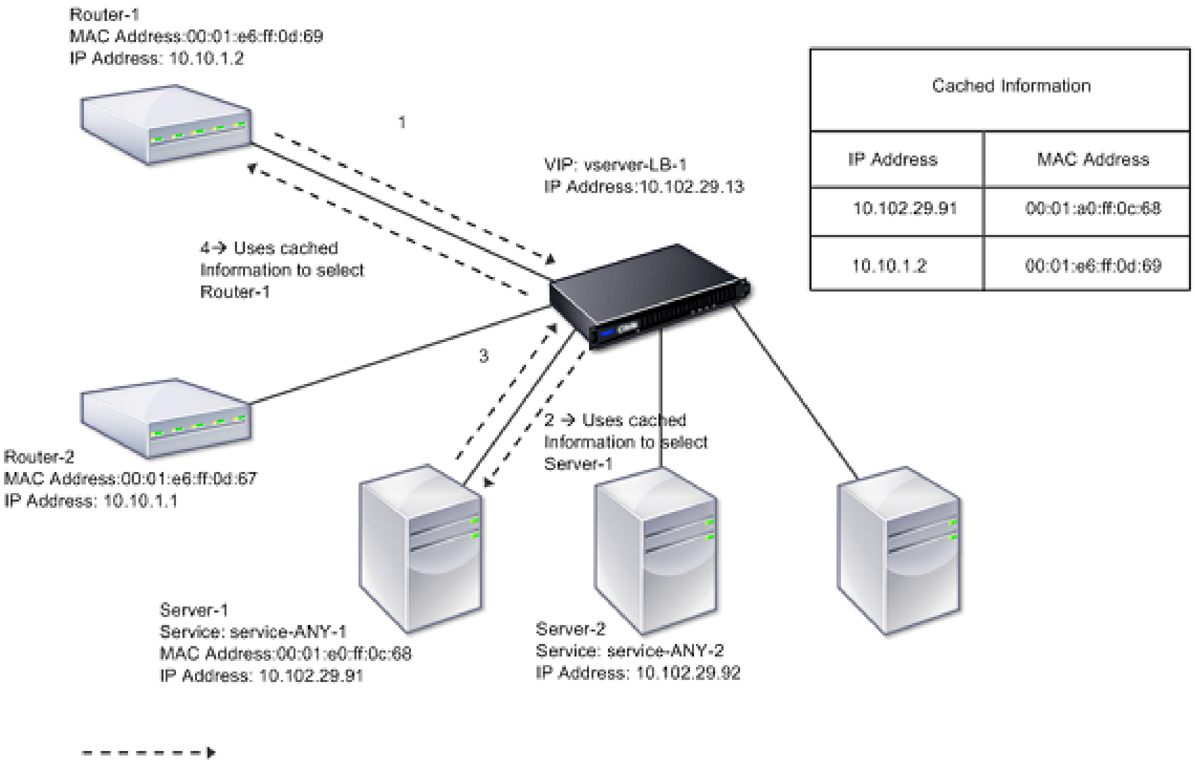
September 21, 2020

With MAC-based forwarding (MBF) enabled, when a request reaches the NetScaler appliance, the appliance remembers the source MAC address of the frame and uses it as the destination MAC address for the resulting replies. MAC-based forwarding can be used to avoid multiple-route/ARP lookups and to avoid asymmetrical packet flows. MAC-based forwarding may be required when the NetScaler is connected to multiple stateful devices, such as VPNs or firewalls, because it ensures that the return traffic is sent to the same device that the initial traffic came from.

MAC-based forwarding is useful when you use VPN devices, because it guarantees that all traffic flowing through a VPN passes back through the same VPN device.

The following topology diagram illustrates the process of MAC-based forwarding.

Figure 1. MAC-Based Forwarding Mode


Router-1
MAC Address:00:01:e6:ff:0d:69
IP Address: 10.10.1.2
1
VIP: vserver-LB-1
IP Address:10.102.29.13
4→ Uses cached Information to select Router-1
3
2 → Uses cached Information to select Server-1
Router-2
MAC Address:00:01:e6:ff:0d:67
IP Address: 10.10.1.1
Server-1
Service: service-ANY-1
MAC Address:00:01:e0:ff:0c:68
IP Address: 10.102.29.91
Server-2
Service: service-ANY-2
IP Address: 10.102.29.92

Cached Information	
IP Address	MAC Address
10.102.29.91	00:01:a0:ff:0c:68
10.10.1.2	00:01:e6:ff:0d:69

When MAC-based forwarding (MBF) is enabled, the NetScaler caches the MAC address of:

- The source (a transmitting device such as router, firewall, or VPN device) of the inbound connection.
- The server that responds to the requests.

When a server replies through the NetScaler appliance, the appliance sets the destination MAC address of the response packet to the cached address, ensuring that the traffic flows in a symmetric manner, and then forwards the response to the client. The process bypasses the route table lookup and ARP lookup functions. However, when the NetScaler initiates a connection, it uses the route and ARP tables for the lookup function. In a direct server return configuration, you must enable MAC-based forwarding.

For more information about direct server return configurations, see Load Balancing.

Some deployment topologies may require the incoming and outgoing paths to flow through different routers. MAC-based forwarding would break this topology design.

MBF should be disabled in the following situations:

- **When you configure link load balancing.** In this case, asymmetric traffic flows are desirable because of link costs.
- **When a server uses network interface card (NIC) teaming without using LACP (802.1ad Link Aggregation).** To enable MAC-based forwarding in this situation, you must use a layer 3 device between the NetScaler and server.
  Note: MBF can be enabled when the server uses NIC teaming with LACP, because the virtual interface uses one MAC address.
- **When firewall clustering is used**. Firewall clustering assumes that ARP is used to resolve the MAC address for inbound traffic. Sometimes the inbound MAC address can be a non-clustered MAC address and should not be used for inbound packet processing.

When MBF is disabled, the NetScaler uses L2 or L3 connectivity to forward the responses from servers to the clients. Depending on the route table, the routers used for outgoing connection and incoming connection can be different. In the case of reverse traffic (response from the server):

- If the source and destination are on different IP subnets, the NetScaler uses the route lookup to locate the destination.
- If the source is on the same subnet as the destination, the NetScaler looks up the ARP table to locate the network interface and forwards the traffic to it. If the ARP table does not exist, the NetScaler requests the ARP entries.

To enable or disable MAC-based forwarding by using the NetScaler command line:

At the command prompt, type:

- **enable ns mode MBF**
- **disable ns mode MBF**

To enable or disable MAC-based forwarding by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in the **Modes and Features** group, click **Configure modes**.
2. Select or clear the **MAC-based forwarding** option.

Configuring Network Interfaces

September 21, 2020

Network interfaces in the NetScaler appliance are numbered in <slot>/<port> notation. After configuring your interfaces, you should display the interfaces and their settings to verify the configuration. You can also display this information to troubleshoot a problem in the configuration.

To manage the network interfaces, you might have to enable some interfaces and disable others. You can reset an interface to renegotiate its settings. You can clear the accumulated statistics for an interface. To verify the configuration, you can display the interface settings. You can display the statistics for an interface to evaluate its health.

Setting the Network Interface Parameters

The network interface configuration is neither synchronized nor propagated. For an HA pair, you must perform the configuration on each unit independently.

To set the network interface parameters by using the NetScaler command line:

At the command prompt, type:

- set interface <id> [-speed <speed>] [-duplex <duplex>] [-flowControl <flowControl>] [-autoneg (DISABLED | ENABLED)] [-haMonitor (ON | OFF)] [(ON | OFF)] [-tagall (ON | OFF)] [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)] [-ifAlias <string>] [-throughput <positive_integer>][-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show interface [<id>]

Example:

```
> set interface 1/8 -duplex full
 Done
<!--NeedCopy-->
```

To set the network interface parameters by using the NetScaler GUI:

Navigate to System > Network > Interfaces, select the network interface that you want to modify (for example, 1/8), click Edit, and then set the parameters.

Enabling and Disabling Network Interfaces

By default, the network interfaces are enabled. You must disable any network interface that is not connected to the network, so that it cannot send or receive packets. Disabling a network interface that is connected to the network in a high availability setup can cause failover.

For more information about high availability, see High Availability.

To enable or disable a network interface by using the NetScaler command line:

At the command prompt, type:

- enable interface <interface_num>
- show interface <interface_num>
- disable interface <interface_num>
- show interface <interface_num>

Example:

```
> enable interface 1/8
 Done
> show interface 1/8
        Interface 1/8 (Gig Ethernet 10/100/1000 MBits) #2
        flags=0x4004000 <ENABLED, DOWN, BOUND to LA/1, down, autoneg,
           802.1q>
        MTU=1514, MAC=00:d0:68:15:fd:3d, downtime 906h58m40s
        Requested: media UTP, speed AUTO, duplex FULL, fctl OFF,
           throughput 0
        RX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        TX: Pkts(0) Bytes(0) Errs(0) Drops(0) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
Done
<!--NeedCopy-->
```

To enable or disable a network interface by using the NetScaler GUI:

1. Navigate to System > Network > Interfaces.
2. Select the network interface and, in the Action list, select Enable or Disable.

Resetting Network Interfaces

Network interface settings control properties such as duplex and speed. To renegotiate the settings of a network interface, you must reset it.

To reset a network interface by using the NetScaler command line:

At the command prompt, type:

- reset interface <interface_num>
- show interface <interface_num>

Example:

```
> reset interface 1/8
 Done
<!--NeedCopy-->
```

To reset a network interface by using the NetScaler GUI:

1. Navigate to System > Network > Interfaces.
2. Select the network interface and, in the Action list, select Reset Interface.

Monitoring a Network Interface

You can display network interface statistics to monitor parameters such as packets sent and packets received, throughput, Link Aggregate Control Protocol (LACP) data units, and errors, and use the information to check the health of the network interface. You can clear the statistics of a network interface to monitor its statistics from the time the statistics are cleared.

To display the statistics of the network interfaces by using the NetScaler command line:

At the command prompt, type:

- **stat interface** <interface_num>

Example:

```
> stat interface 1/8
 Done
<!--NeedCopy-->
```

To clear a network interface's statistics by using the NetScaler command line:

At the command prompt, type:

- **clear interface** <interface_num>

Example:

```
> clear interface 1/8
 Done
<!--NeedCopy-->
```

To display the statistics of an Interface by using the NetScaler GUI:

Navigate to System > Network > Interfaces, select the network interface, and click Interface Statistics.

To clear a network interface's statistics by using the NetScaler GUI:

1. Navigate to System > Network > Interfaces.
2. Select the network interface and, in the Action list, select Clear Statistics.

Configuring Forwarding Session Rules

September 21, 2020

By default, the NetScaler appliance does not create session entries for traffic that it only forwards (L3 mode). For a case in which a client request that the appliance forwards to a server results in a response that has to return by the same path, you can create a forwarding-session rule. A forwarding-session rule creates forwarding-session entries for traffic that originates from or is destined for a particular network and is forwarded by the NetScaler. You can create forwarding session rules for IPv4 traffic as well as IPv6 traffic.

When configuring an IPv4 forwarding-session rule, you can specify either an IPv4 network address or an extended ACL as the condition for identifying IPv4 traffic for which to create a forwarding-session entry:

- **Network address**. When you specify an IPv4 network address, the appliance creates forwarding sessions for IPv4 traffic whose source or destination matches the network address.
- **Extended ACL rule**. When you specify an extended ACL rule, the appliance creates forwarding sessions for IPv4 traffic that matches the conditions specified in the extended ACL rule.

When configuring an IPv6 forwarding-session rule, you can specify either an IPv6 prefix or an ACL6 as the condition for identifying IPv6 traffic for which to create a forwarding-session entry:

- **IPv6 prefix**. When you specify an IPv6 prefix, the appliance creates forwarding sessions for IPv6 traffic whose source or destination matches the IPv6 prefix.
- **ACL6 rule**. When you specify an ACL6 rule, the appliance creates forwarding sessions for IPv6 traffic that matches the conditions specified in the ACL6 rule.

To create an IPv4 forwarding session rule by using the NetScaler command line:

At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:

- add forwardingSession <name> [<network> <netmask>] | [-aclname <string>] -connfailover (ENABLED | DISABLED)

- show forwardingSession

Example:

```
A network address as the condition:

> add forwardingSession fs-nw-1 10.102.105.51 255.255.255.255
 Done

An ACL as the condition:

> add forwardingSession fs-acl-1 acl1
 Done
<!--NeedCopy-->
```

To configure an IPv4 forwarding session rule by using the NetScaler GUI:

Navigate to System > Network > Forwarding Sessions, add a new IPv4 forwarding session, or edit an existing forwarding session.

To create an IPv6 forwarding session rule by using the NetScaler command line:

- At the command prompt, type the following commands to create a forwarding-session rule and verify the configuration:
  - add forwardingSession <name> [<IPv6 prefix>] | [-acl6name <string>]
  - show forwardingSession

  Example:

```
     An IPv6 prefix as the condition:

     > add forwardingSession fsv6-pfx-1 3ffe::/64
      Done

     An ACL6 rule as the condition:

     > add forwardingSession fsv6-acl6-1  - acl6name ACL6-FS
      Done
<!--NeedCopy-->
```

To configure an IPv6 forwarding session rule by using the NetScaler GUI:

Navigate to System > Network > Forwarding Sessions, add a new IPv6 forwarding session, or edit an existing forwarding session.

Assigning an ACL rule to an Existing Forwarding Session Rule

You can assign an ACL rule to a Network-address/IPv6-prefix based forwarding session rule, in which case it becomes an ACL based forwarding session rule. You can also change an existing ACL rule to another ACL rule in an ACL based forwarding session rule. After the existing related forwarding session entries (if any) have timed out, the rules start using the newly assigned ACL to identify IPv4/IPv6 traffic for which to create a forwarding-session entry.

To assign an extended ACL rule to an existing IPv4 forwarding session rule by using the NetScaler command line:

At the command prompt, type

- **set forwardingSession** <name> [-**aclname** <string>]
- **show forwardingSession** <name>

To assign an ACL6 rule to an existing IPv6 forwarding session rule by using the NetScaler command line:

At the command prompt, type

- **set forwardingSession** <name> [-**acl6name** <string>]

- **show forwardingSession** <name>

Example:

```
> add forwardingSession FS-1 -aclname ACL-9
Done

> add forwardingSession FS6-1  – acl6name ACL6-9
Done
```

Disabling Steering for Forwarding Sessions on a Cluster Setup

The default behavior of a NetScaler cluster is for the node that receives traffic (flow receiver) to direct the traffic to another node (flow processor), which processes the traffic. Directing the traffic from flow receiver to flow processor occurs over the cluster backplane and is called steering.

Steering can be an overhead for real-time processing or when the setup includes high-latency links.

Steering for forwarding sessions can now be disabled so that the processing becomes local to the flow receiver. That is, the flow receiver becomes the flow processor.

Before you begin

Note the following points before configuring forwarding session rules in a cluster setup:

- You must configure linksets to be used for forwarding sessions.
- You must enable MAC Based Forwarding (MBF) on the cluster setup.

Configuring Forwarding Session Rules in a Cluster Setup

Disabling steering for forwarding session rules in a cluster setup can be done at the following two levels:

- **Specific forwarding session rule level**. Enable the Process Local parameter while adding a new forwarding session rule or editing an existing forwarding session rule.
- **Global level**. Enable the Process Local parameter while adding a new cluster instance or editing an existing cluster instance. The global setting takes precedence over the forwarding session rule setting.

NetScaler command line procedures

To disable steering for a forwarding session rule on a cluster setup by using the NetScaler command line:

At the command prompt, type one of the following sets of commands:

- If adding a new forwarding session rule:
  - **add forwardingSession** <name> ((<network> [<netmask>]) | -**acl6name** <string> | -**aclname** <string>) -**processLocal ENABLED**
  - **show forwardingSession** <name>
- If reconfiguring an existing forwarding session rule:
  - **set forwardingSession** <name> -**processLocal ENABLED**
  - **show forwardingSession** <name>

To disable steering for all (global level) forwarding session rules on a cluster setup by using the NetScaler command line:

At the command prompt, type one of the following sets of commands:

- If adding a new cluster instance:
  - **add cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>
- If reconfiguring an existing cluster instance:
  - **set cluster instance** <clid> -**processLocal Enabled**
  - **show cluster instance** <clid>

Sample configuration:

Following are two examples of disabling steering at the forwarding session rule level, and an example of disabling steering at the global level.

```
An IPv4 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV4-1 10.102.105.51 255.255.255.255 -processLocal Enabled
Done

An IPv6 forwarding session rule:

> add forwardingSession FWD-SESSN-PROCSS-LOCL-IPV6-1  - acl6name ACL6-FWD-SESSN-1 -processLocal Enabled
Done

A cluster setup, with an instance ID 10, has steering disabled at global level:

> set cluster instance 10 -processLocal Enabled
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To disable steering for a forwarding session rule on a cluster setup by using the NetScaler GUI:

Navigate to **System** > **Network** > **Forwarding Sessions**, select **Process Local** while adding a new forwarding session rule or editing an existing forwarding session rule.

To disable steering for all (global level) forwarding session rules on a cluster setup by using the NetScaler GUI:

Navigate to **System** > **Cluster**, and select **Process Local** while adding a cluster configuration or modifying an existing cluster configuration.

Understanding VLANs

September 21, 2020

A NetScaler appliance supports Layer 2 port and IEEE 802.1q tagged VLANs. VLAN configurations are useful when you need to restrict traffic to certain groups of stations. You can configure a network interface as a part of multiple VLANs by using IEEE 802.1q tagging.

You can configure VLANs and bind them to IP subnets. The NetScaler then performs IP forwarding between these VLANs (if it is configured as the default router for the hosts on these subnets).

The NetScaler supports the following types of VLANs:

- **Port-Based VLANs.** The membership of a port-based VLAN is defined by a set of network interfaces that share a common, exclusive Layer 2 broadcast domain. You can configure multiple port-based VLANs. By default, all network interfaces on the NetScaler are members of VLAN 1.

  If you apply 802.1q tagging to the port, the network interface belongs to a port-based VLAN. Layer 2 traffic is bridged within a port-based VLAN, and Layer 2 broadcasts are sent to all members of the VLAN if Layer 2 mode is enabled. When you add an untagged network interface as a member of a new VLAN, it is removed from its current VLAN.

- **Default VLAN.** By default, the network interfaces on the NetScaler are included in a single, port-based VLAN as untagged network interfaces. This VLAN is the default VLAN. It has a VLAN ID (VID) of 1. This VLAN exists permanently. It cannot be deleted, and its VID cannot be changed.

  When you add a network interface to a to a different VLAN as an untagged member, the network interface is automatically removed from the default VLAN. If you unbind a network interface from its current port-based VLAN, it is added to the default VLAN again.

- **Tagged VLANs.** 802.1q tagging (defined in the IEEE 802.1q standard) allows a networking device (such as the NetScaler) to add information to a frame at Layer 2 to identify the VLAN membership of the frame. Tagging allows network environments to have VLANs that span multiple devices.

A device that receives the packet reads the tag and recognizes the VLAN to which the frame belongs. Some network devices do not support receiving both tagged and untagged packets on the same network interface—in particular, Force10 switches. In such cases, you need to contact customer support for assistance.

The network interface can be a tagged or untagged member of a VLAN. Each network interface is an untagged member of one VLAN only (its native VLAN). This network interface transmits the frames for the native VLAN as untagged frames. A network interface can be a part of more than one VLAN if the other VLANs are tagged.

When you configure tagging, be sure to match the configuration of the VLAN on both ends of the link. The port to which the NetScaler connects must be on the same VLAN as the NetScaler network interface.

Note: This VLAN configuration is neither synchronized nor propagated, therefore you must perform the configuration on each unit in an HA pair independently.

Applying Rules to Classify Frames

VLANs have two types of rules for classifying frames:

- **Ingress rules.** Ingress rules classify each frame as belonging only to a single VLAN. When a frame is received on a network interface, the following rules are applied to classify the frame:
  - If the frame is untagged, or has a tag value equal to 0, the VID of the frame is set to the port VID (PVID) of the receiving interface, which is classified as belonging to the native VLAN. (PVIDs are defined in the IEEE 802.1q standard.)
  - If frame has a tag value equal to FFF, the frame is dropped.
  - If the VID of the frame specifies a VLAN of which the receiving network interface is not a member, the frame is dropped. For example, if a packet is sent from a subnet associated with VLAN ID 12 to a subnet associated with VLAN ID 10, the packet is dropped. If an untagged packet with VID 9 is sent from the subnet associated with VLAN ID 10 to a network interface PVID 9, the packet is dropped.
- **Egress Rules.** The following egress rules are applied:
  - If the VID of the frame specifies a VLAN of which the transmission network interface is not a member, the frame is discarded.
  - During the learning process (defined by the IEEE 802.1q standard), the Src MAC and VID are used to update the bridge lookup table of the NetScaler.
  - A frame is discarded if its VID specifies a VLAN that does not have any members. (You define members by binding network interfaces to a VLAN.)

VLANs and Packet Forwarding on the NetScaler

The forwarding process on the NetScaler appliance is similar to that on any standard switch. However, the NetScaler performs forwarding only when Layer 2 mode is on. The key features of the forwarding process are:

- Topology restrictions are enforced. Enforcement involves selecting each network interface in the VLAN as a transmission port (depending on the state of the network interface), bridging restrictions (do not forward on the receiving network interface), and MTU restrictions.
- Frames are filtered on the basis of information in the bridge table lookup in the forwarding database (FDB) table of the NetScaler. The bridge table lookup is based on the destination MAC and the VID. Packets addressed to the MAC address of the NetScaler are processed at the upper layers.
- All broadcast and multicast frames are forwarded to each network interface that is a member of the VLAN, but forwarding occurs only if L2 mode is enabled. If L2 mode is disabled, the broadcast and multicast packets are dropped. This is also true for MAC addresses that are not currently in the bridging table.
- A VLAN entry has a list of member network interfaces that are part of its untagged member set. When forwarding frames to these network interfaces, a tag is not inserted in the frame.
- If the network interface is a tagged member of this VLAN, the tag is inserted in the frame when the frame is forwarded.

When a user sends any broadcast or multicast packets without the VLAN being identified, that is, during duplicate address detection (DAD) for NSIP or ND6 for the next hop of the route, the packet is sent out on all the network interfaces, with appropriate tagging based on either the Ingress and Egress rules. ND6 usually identifies a VLAN, and a data packet is sent on this VLAN only. Port-based VLANs are common to IPv4 and IPv6. For IPv6, the NetScaler supports prefix-based VLANs.

Configuring a VLAN

September 21, 2020

You can implement VLANs in the following environments:

- Single subnet
- Multiple subnets
- Single LAN
- VLANs (no tagging)
- VLANs (802.1q tagging)

If you configure VLANs that have only untagged network interfaces as their members, the total number of possible VLANs is limited to the number of network interfaces available in the NetScaler. If more IP subnets are required with a VLAN configuration, 802.1q tagging must be used.

When you bind a network interface to a VLAN, the network interface is removed from the default VLAN. If the network interfaces need to be a part of more than one VLAN, you can bind the network interfaces to the VLANs as tagged members.

You can configure the NetScaler to forward traffic between VLANs at Layer 3. In this case, a VLAN is associated with a single IP subnet. The hosts in a VLAN that belong to a single subnet use the same subnet mask and one or more default gateways connected to that subnet. Configuring Layer 3 for a VLAN is optional. Layer 3 is used for IP forwarding (inter-VLAN routing). Each VLAN has a unique IP address and subnet mask that define an IP subnet for the VLAN. In an HA configuration, this IP address is shared with the other NetScaler appliances. The NetScaler forwards packets between configured IP subnets (VLANs).

When you configure the NetScaler, you must not create overlapping IP subnets. Doing so impedes Layer 3 functionality.

Each VLAN is a unique Layer 2 broadcast domain. Two VLANs, each bound to separate IP subnets, cannot be combined into a single broadcast domain. Forwarding traffic between two VLANs requires a Layer 3 forwarding (routing) device, such as the NetScaler appliance.

Configuring VLANs in an HA Setup

VLAN configuration for a high-availability setup requires that the NetScaler appliances have the same hardware configuration, and the VLANs configured on them must be mirror images.

The correct VLAN configuration is implemented automatically when the configuration is synchronized between the NetScaler appliances. The result is identical actions on all the appliances. For example, adding network interface 0/1 to VLAN2 adds this network interface to VLAN 2 on all the appliances participating in the high-availability setup.

Note: If you use network-interface-specific commands in an HA setup, the configurations you create are not propagated to the other NetScaler appliance. You must perform these commands on each appliance in an HA pair to ensure that the configuration of the two appliances in the HA pair remains synchronized.

Creating or Modifying a VLAN

To configure a VLAN, you create a VLAN entity, and then bind network interfaces and IP addresses to the VLAN. If you remove a VLAN, its member interfaces are added to the default VLAN.

NetScaler command line procedures

To create a VLAN by using the NetScaler command line:

At the command prompt, type:

- add vlan <id> [-aliasName <string>] [-ipv6DynamicRouting (ENABLED|DISABLED)]
- sh vlan <id>

Example:

```
  > add vlan 2  - aliasName  “Network A”    Done
<!--NeedCopy-->
```

To bind an interface to a VLAN by using the NetScaler command line:

At the command prompt, type:

- bind vlan <id> -ifnum <slot/port>
- sh vlan <id>

Example:

```
  > bind vlan 2 -ifnum 1/8   Done
<!--NeedCopy-->
```

To bind an IP address to a VLAN by using the NetScaler command line:

At the command prompt, type:

- bind vlan <id> -IPAddress <IPAddress> <netMask>
- sh vlan <id>

Example:

```
  > bind vlan 2 -IPAddress 10.102.29.54 255.255.255.0   Done
<!--NeedCopy-->
```

To remove a VLAN by using the NetScaler command line:

At the command prompt, type:

- rm vlan <id>

NetScaler GUI procedures

To configure a VLAN by using the NetScaler GUI:

1. Navigate to System > Network > VLANs, add a new VLAN, or edit an existing VLAN.
2. To bind an IP address to a VLAN, under IP Bindings, select the Active option corresponding to the IP address that you want to bind to the VLAN (for example, 10.102.29.54). The Type column displays the IP address type (such as mapped IP, virtual IP, or subnet IP) for each IP address in the IP Address column.

3. To bind a network interface to a VLAN, under Interface Bindings, select the Active option corresponding to the interface that you want to bind to the VLAN.

Monitoring VLANs

You can display VLAN statistics such as packets received, bytes received, packets sent, and bytes sent, and use the information to identify anomalies and or debug a VLAN.

To view the statistics of a VLAN by using the NetScaler command line:

At the command prompt, type:

- stat vlan <vlanID>

Example:

```
  stat vlan 2
<!--NeedCopy-->
```

To view the statistics of a VLAN by using the NetScaler GUI:

1. Navigate to System > Network > VLANs.
2. Select the VLAN, and click Statistics.

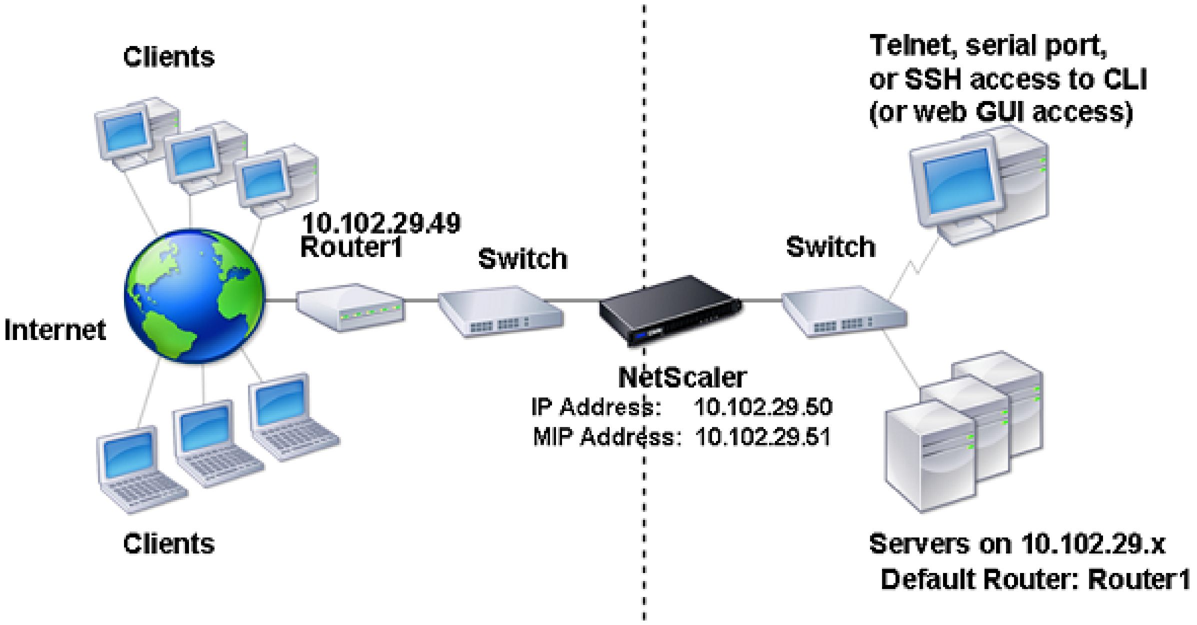
Configuring VLANs on a Single Subnet

September 21, 2020

Before configuring a VLAN on a single subnet, make sure that Layer 2 Mode is enabled.

The following figure shows a single subnet environment

Figure 1. VLAN on a Single Subnet


Clients
Internet
Clients
10.102.29.49
Router1
Switch
NetScaler
IP Address: 10.102.29.50
MIP Address: 10.102.29.51
Switch
Telnet, serial port,
or SSH access to CLI
(or web GUI access)
Servers on 10.102.29.x
Default Router: Router1

In the above figure:

1. The default router for the NetScaler and the servers is Router 1.
2. Layer 2 mode must be enabled on the NetScaler for the NetScaler to have direct access to the servers.
3. For this subnet, a virtual server can be configured for load balancing on the NetScaler appliance.

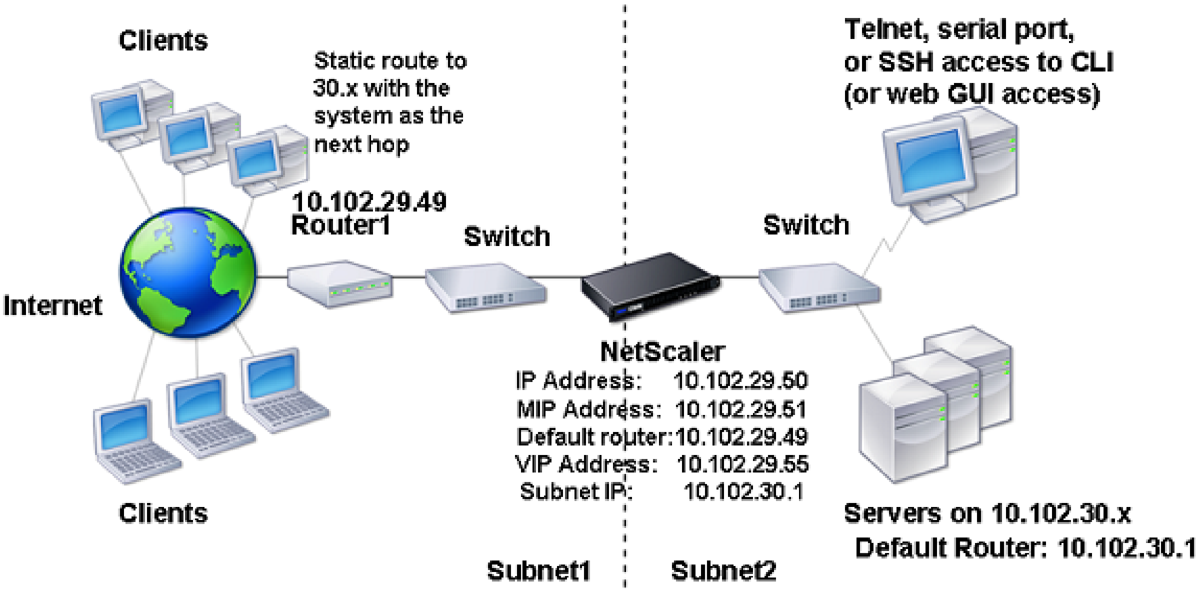
To configure a VLAN on a single subnet, follow the procedures described in Configuring a VLAN.

Configuring VLANs on Multiple Subnets

September 21, 2020

To configure a single VLAN across multiple subnets, you must add a VIP for the VLAN and configure the routing appropriately. The following figure shows a single VLAN configured across multiple subnets.

Figure 1. Multiple Subnets in a Single VLAN


Clients
Static route to 30.x with the system as the next hop
10.102.29.49
Router1
Switch
Internet
Clients
Telnet, serial port, or SSH access to CLI (or web GUI access)
Switch
NetScaler
IP Address: 10.102.29.50
MIP Address: 10.102.29.51
Default router:10.102.29.49
VIP Address: 10.102.29.55
Subnet IP: 10.102.30.1
Servers on 10.102.30.x
Default Router: 10.102.30.1
Subnet1
Subnet2

To configure a single VLAN across multiple subnets, perform the following tasks:

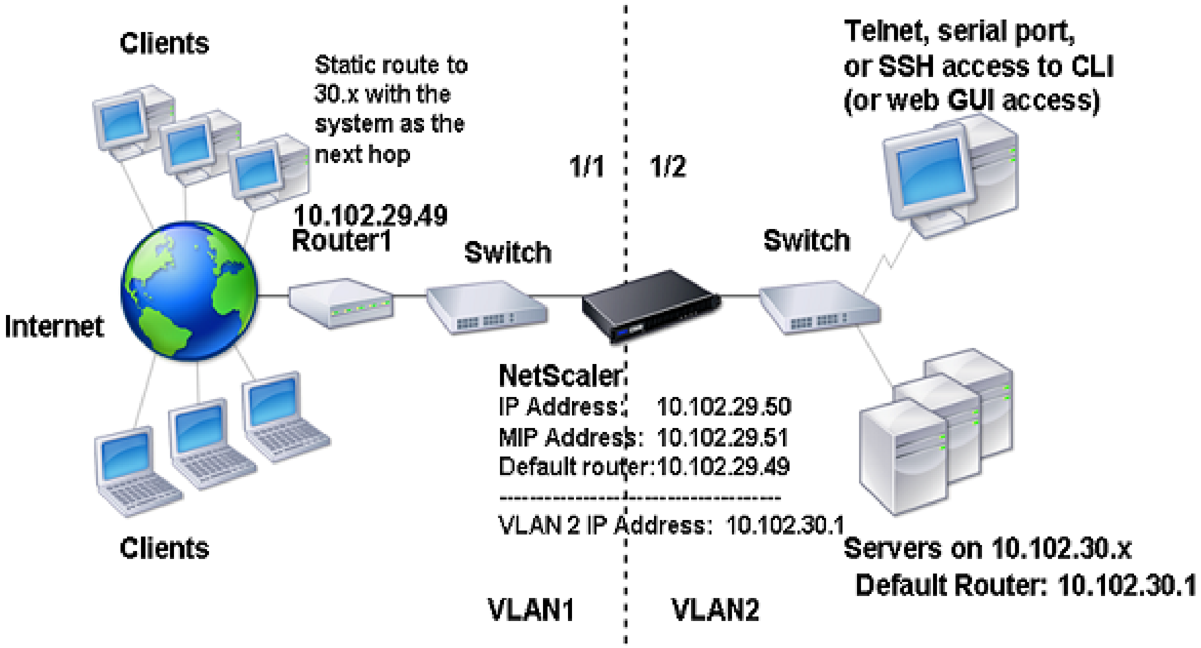
1. Disable Layer 2 mode. For the procedure to disable Layer 2 mode, see Packet forwarding modes.
2. Add a VIP address. For the procedure to add a VIP address, see Configuring and Managing Virtual IP Addresses (VIPs).
3. Configure RNAT rule. For the procedure to configure the RNAT ID, see Configuring RNAT.

Configuring Multiple Untagged VLANs across Multiple Subnets

September 21, 2020

In environments with multiple untagged VLANs across multiple subnets, a VLAN is configured for each IP subnet. A network interface is bound to one VLAN only. The following figure shows this configuration.

Figure 1. Multiple Subnets with VLANs - No Tagging


Clients
Static route to 30.x with the system as the next hop
1/1
1/2
Telnet, serial port, or SSH access to CLI (or web GUI access)
10.102.29.49
Router1
Switch
Switch
Internet
NetScaler
IP Address: 10.102.29.50
MIP Address: 10.102.29.51
Default router:10.102.29.49
VLAN 2 IP Address: 10.102.30.1
Clients
Servers on 10.102.30.x
Default Router: 10.102.30.1
VLAN1
VLAN2

To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.
2. Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.
3. Bind the IP address and subnet mask to VLAN 2.

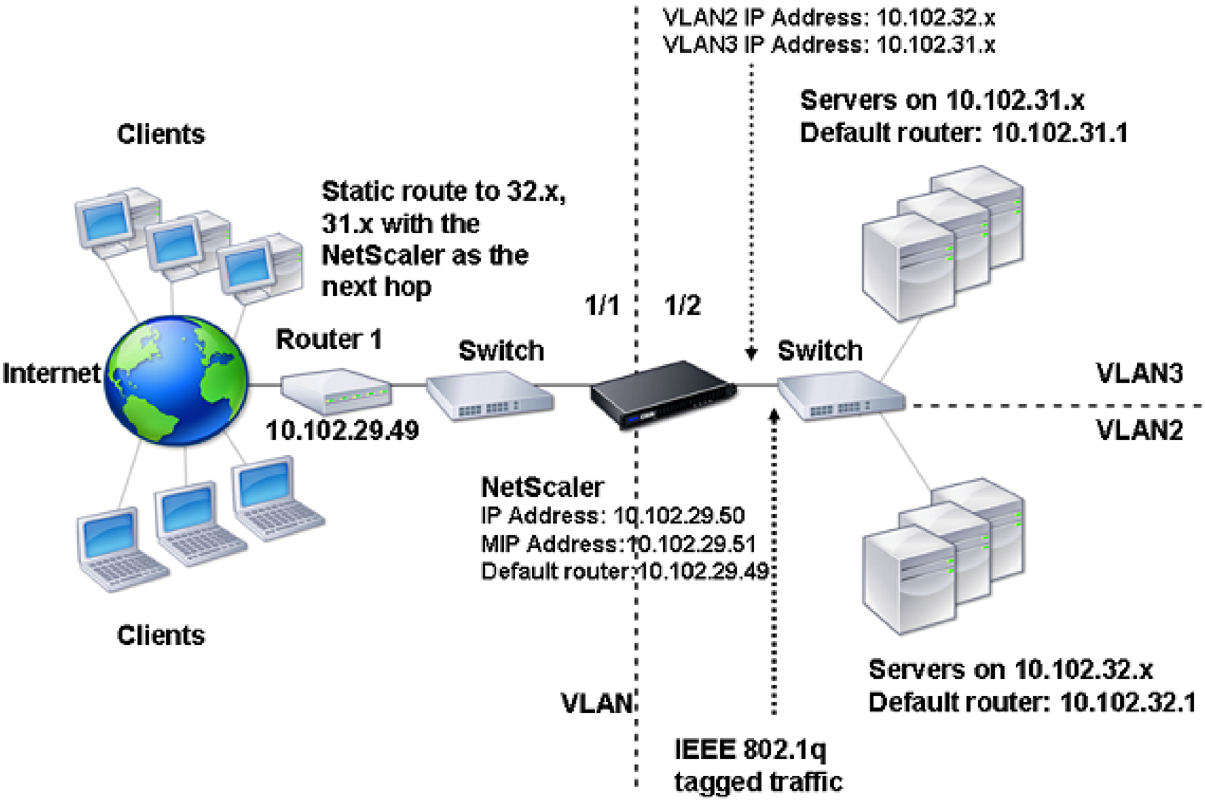
For procedures on these tasks, see Configuring a VLAN.

Configuring Multiple VLANs with 802.1q Tagging

September 21, 2020

For multiple VLANs with 802.1q tagging, each VLAN is configured with a different IP subnet. Each network interface is in one VLAN. One of the VLANs is set up as tagged. The following figure shows this configuration.

Figure 1. Multiple VLANs with IEEE 802.1q Tagging


VLAN2 IP Address: 10.102.32.x
VLAN3 IP Address: 10.102.31.x
Servers on 10.102.31.x
Default router: 10.102.31.1
Clients
Static route to 32.x, 31.x with the NetScaler as the next hop
1/1
1/2
Router 1
Switch
Switch
Internet
VLAN3
VLAN2
10.102.29.49
NetScaler
IP Address: 10.102.29.50
MIP Address: 10.102.29.51
Default router: 10.102.29.49
Clients
Servers on 10.102.32.x
Default router: 10.102.32.1
VLAN
IEEE 802.1q tagged traffic

To implement the configuration shown in the above figure, perform the following tasks:

1. Add VLAN 2.
2. Bind the 1/2 network interface of the NetScaler to VLAN 2 as an untagged network interface.
3. Bind the IP address and netmask to VLAN 2.
4. Add VLAN 3.
5. Bind the 1/2 network interface of the NetScaler to VLAN 3 as a tagged network interface.
6. Bind the IP address and netmask to VLAN 3.

For procedures on these tasks, see Configuring a VLAN.

Configuring NSVLAN

September 21, 2020

NSVLAN is a VLAN to which the NetScaler management IP (NSIP) address's subnet is bound. The NSIP subnet is available only on interfaces that are associated with NSVLAN. By default, NSVLAN is VLAN 1, but you can designate a different VLAN as NSVLAN. If you do so, you must reboot the NetScaler appliance for the change to take effect. After the reboot, NSIP subnet traffic is restricted to the new NSVLAN.

The traffic from the NetScaler IP subnet can be tagged (802.1q) with the VLAN ID specified for NSVLAN. You must configure the attached switch interface to tag and allow this same VLAN ID on the connected interface. If you remove your NSVLAN configuration, the NSIP subnet is automatically bound to VLAN 1, restoring the default NSVLAN.

To configure NSVLAN by using the NetScaler command line:

At the command prompt, type:

- set ns config -nsvlan <positive_integer> -ifnum <interface_name> … [-tagged (YES|NO)]
- show ns config

  Note: The configuration takes effect after the NetScaler appliance is rebooted.

Example:

```
> set ns config -nsvlan 300 -ifnum 1/1 1/2 1/3 -tagged NO
 Done

> save config
 Done
<!--NeedCopy-->
```

To restore the default NSVLAN configuration by using the NetScaler command line:

At the command prompt, type:

- unset ns config -nsvlan
- show ns config

Example:

```
> unset ns config -nsvlan
 Done
<!--NeedCopy-->
```

To configure NSVLAN by using the NetScaler GUI:

Navigate to System > Settings, in the Settings group, click Change NSVLAN Settings.

Setting the MTU on the NSVLAN

By default, the MTU of the NSVLAN is set to 1500 bytes. You can modify this setting to optimize throughput and network performance. For example, you can configure the NSVLAN to process jumbo frames.

To set the MTU of the NSVLAN by using the NetScaler command line:

At the command prompt, type:

- **set vlan** <id> **-mtu** <positive_integer>
- **show vlan** <id>

To set the MTU of the NSVLAN by using the NetScaler GUI:

Navigate to **System** > **Network** > **VLANs**, open the NSVLAN, and set the **Maximum Transmission Unit** parameter.

Sample configuration:

In the following sample configuration, VLAN 100 is the NSVLAN.

```
> set ns config -nsvlan 100 -ifnum 1/1 -tagged no

  Warning: The configuration must be saved and the system rebooted for
      these settings to take effect

> set vlan 100 -mtu 1600

  Done

> sh vlan

  1)  VLAN ID: 1

      Link-local IPv6 addr:
    fe80::947b:52ff:fead:12d5/64

      Interfaces : 1/2 LO/1

  2)  VLAN ID: 100    VLAN Alias Name:

      MTU: 1600

      Interfaces : 1/1

      IPs :

          10.102.53.114     Mask: 255.255.255.0

  Done

> save config

  Done
<!--NeedCopy-->
```

Configuring Allowed VLAN List

September 21, 2020

NetScaler accepts and sends tagged packets of a VLAN on an interface if the VLAN is explicitly configured on the NetScaler appliance and the interface is bound to the VLAN. Some deployments (for example, Bump in the wire) require the NetScaler appliance to function as a transparent device to accept and forward tagged packets related to a large number of VLANs. For this requirement, configuring and managing a large number of VLANs is not a feasible solution.

Allowed VLAN list on an interface specifies a list of VLANs. The interface transparently accepts and sends tagged packets related to the specified VLANs without the need for explicitly configuring these VLANs on the appliance.

Points to Consider before Configuring Allowed VLAN List

Consider the following points before configuring allowed VLAN list

- In a high availability setup, allowed VLAN list is not propagated or synchronized. Therefore, you have to configure allowed VLAN list on both the nodes.
- The traffic of a native VLAN might leak to the non-member interfaces that specifies the native VLAN in its allowed VLAN list.
- A Maximum of 60 VLAN ranges can be specified as part of allowed VLAN list for an interface.
- The NetScaler appliance does not support allowed VLAN list on interfaces that are part of link aggregation channels or redundant interface sets. For more information on redundant interface set, see Redundant Interface Set.
- Allowed VLAN list is not supported on a NetScaler cluster configuration.
- The NetScaler appliance does not support allowed VLAN list for Bridge groups.
- The NetScaler appliance does not support allowed VLAN list for VXLANs.

Configuring Allowed VLAN List

To configure allowed VLAN list by using the NetScaler command line:

At the command prompt, type:

- **set interface** <id> **-trunkmode** (ON|OFF) **-trunkAllowedVlan** <int[-int]> …
- **show interface** <id>

To configure allowed VLAN list by using the NetScaler GUI:

Navigate to **System** > **Network** > **Interfaces**, select a network interface, click **Edit**, and then set the following parameters:

- Trunk Mode
- Trunk Allowed VLAN

Sample Configuration:

In the following sample configuration, VLANS in the ranges 100-120, 190-200, and 300-330 are specified as part of allowed VLAN list for interface 1/2.

```
> set int 1/2 -trunkmode on -trunkallowedVlan 100-120 190-200 300-330

Done

> sh int 1/2

1)      Interface 1/2 (Gig Ethernet 10/100/1000 MBits) #6
        flags=0xc020

        <ENABLED, UP, UP, AUTONEG OFF, HEARTBEAT, 802.1q, trunkmode>

        Trunk Allowed Vlans:  100-120 190-200 300-330

Done
<!--NeedCopy-->
```

Configuring Bridge Groups

September 21, 2020

Typically, when you want to merge two or more VLANs into a single domain, you change the VLAN configuration on all the devices in the separate domains. This can be a tedious task. To more easily merge multiple VLANs into a single broadcast domain, you can use bridge groups.

The bridge groups feature works the same way as a VLAN. Multiple VLANS can be bound to a single bridge group, and all VLANs bound to same bridge group form a single broadcast domain. You can bind only Layer 2 VLANs to a bridge group. For Layer 3 functionality, you must assign an IP address to a bridge group.

In Layer 2 mode, a broadcast packet received on an interface belonging to a particular VLAN is bridged to other VLANs that belong to the same bridge group. In the case of a unicast packet, the NetScaler appliance searches its bridge table for the learned MAC addresses of all the VLANs belonging to same bridge group.

In Layer 3 forwarding mode, an IP subnet is bound to a bridge group. The NetScaler accepts incoming packets belonging to the bound subnet and forwards the packets only on VLANs that are bound to the bridge group.

IPv6 routing can be enabled on a configured bridge group.

> **Note**
>
> Bridge Group feature and Bridge BPDU mode cannot work together.

Configuration steps

Perform the following steps to configure a bridge group:

- Enable Layer 2 mode
- Add a bridgegroup and bind VLANs to the bridgegroup

CLI procedures

To enable Layer 2 mode by using the NetScaler command line:
At the command prompt, type:

- **enable ns mode l2**
- **show ns mode**

To add a bridge group and bind VLANs by using the NetScaler command line:

At the command prompt, type:

- **add bridgegroup** <id> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **bind bridgegroup** <id> -**vlan** <positive_integer>
- **show bridgegroup** <id>

Example:

```
> add bridgegroup 12
 Done
<!--NeedCopy-->
```

To remove a bridge group by using the NetScaler command line:

At the command prompt, type:

- **rm bridgegroup** <id>

Example:

```
 rm bridgegroup 12
<!--NeedCopy-->
```

GUI Procedures

To enable Layer 2 mode by using the NetScaler command line:
At the command prompt, type:

- **enable ns mode l2**
- **show ns mode**

To configure a bridge group by using the NetScaler GUI:

Navigate to **System** > **Network** > **Bridge Groups**, add a new bridge group and bind VLANs to the bridgegroup, or edit an existing bridge group.

Configuring Virtual MACs

September 21, 2020

The primary and secondary nodes in a high availability (HA) setup share the Virtual MAC address floating entity. The primary node owns the floating IP addresses (such as MIP, SNIP, and VIP) and responds to ARP requests for these IP addresses with its own MAC address. Therefore, the ARP table of an external device, such as an upstream router, is updated with the floating IP address and the MAC address of the primary node.

When a failover occurs, the secondary node takes over as the new primary node. The former secondary node uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it had learned from the old primary node. The MAC address that the new primary node advertises is the MAC address of its own network interface. Some devices (a few routers) do not accept these GARP messages. Therefore, these external devices retain the IP address-to-MAC address mapping that the old primary node had advertised. This can result in a GSLB site going down.

Therefore, you must configure a virtual MAC on both nodes of an HA pair. This means that both nodes have identical MAC addresses. When a failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

For the procedures to configure a Virtual MAC, see Configuring Virtual MAC Addresses.

Configuring Link Aggregation

September 21, 2020

Link aggregation combines data coming from multiple ports into a single high-speed link. Configuring link aggregation increases the capacity and availability of the communication channel between the

NetScaler appliance and other connected devices. An aggregated link is also referred to as a "channel." You can configure the channels manually, or you can use Link Aggregation Control Protocol (LACP). You cannot apply LACP to a manually configured channel, nor can you manually configure a channel created by LACP.

When a network interface is bound to a channel, the channel parameters have precedence over the network interface parameters. (That is, the network interface parameters are ignored.) A network interface can be bound only to one channel.

When a network interface is bound to a channel, it drops its VLAN configuration. When network interfaces are bound to a channel, either manually or by LACP, they are removed from the VLANs that they originally belonged to and added to the default VLAN. However, you can bind the channel back to the old VLAN, or to a new one. For example, if you bind the network interfaces 1/2 and 1/3 to a VLAN with ID 2, and then bind them to a channel LA/1, the network interfaces are moved to the default VLAN, but you can bind them back to VLAN 2.

Configuring Link Aggregation Manually

When you create a link aggregation channel, its state is DOWN until you bind an active interface to it. You can modify a channel at any time. You can remove channels, or you can enable/disable them.

NetScaler command line procedures

To create a link aggregation channel by using the NetScaler command line:

At the command prompt, type:

- add channel <id> [-ifnum <interfaceName> …] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)][-tagall (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show channel

Example:

```
> add channel LA/1 -ifnum 1/8
Done
<!--NeedCopy-->
```

To bind an interface to or unbind an interface from an existing link aggregation channel by using the NetScaler command line:

At the command prompt, type one of the following commands:

- bind channel <id> <interfaceName>

- unbind channel <id> <interfaceName>

Example:

```
bind channel LA/1 1/8
<!--NeedCopy-->
```

To modify a link aggregation channel by using the NetScaler command line:

At the command prompt, type the
set channel command, the channel ID, and the parameters to be changed, with their new values.

To remove a link aggregation channel by using the NetScaler command line:

Important: When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

At the command prompt, type:

- rm channel <id>

Example:

```
> rm channel LA/1
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To configure a link aggregation channel by using the NetScaler GUI:

Navigate to System > Network > Channels, add a new channel, or edit an existing channel.

To remove a link aggregation channel by using the NetScaler GUI:

> **Important:**
>
> When a channel is removed, the network interfaces bound to it induce network loops that decrease network performance. You must disable the network interfaces before you remove the channel.

Navigate to System > Network > Channels, select the channel that you want to remove and click Delete.

Configuring Link Aggregation by Using the Link Aggregation Control Protocol

The Link Aggregation Control Protocol (LACP) enables network devices to exchange link aggregation information by exchanging LACP Data Units (LACPDUs). Therefore, you cannot enable LACP on network interfaces that are members of a channel that you created manually.

When using LACP to configure link aggregation, you use different commands and parameters for modifying link aggregation channels than you do for creating link aggregation channels. To remove a channel, you must disable LACP on all interfaces that are part of the channel.

Note: In an High Availability configuration, LACP configurations are neither propagated nor synchronized.

Configuring the LACP System Priority

The LACP system priority determines which peer device of an LACP LA channel can have control over the LA channel. This number is globally applied to all LACP channels on the appliance. The lower the value, the higher the priority.

To configure the LACP system priority by using the NetScaler command line:

At the command prompt, type the following commands to set the priority for a standalone appliance and verify the configuration:

- set lacp -sysPriority <positive_integer>
- show lacp

Example:

```
set lacp -sysPriority 50
<!--NeedCopy-->
```

To set the priority for a specific cluster node, log on to the cluster IP address and at the command prompt, type the following commands:

- set lacp -sysPriority <positive_integer> -ownerNode <positive_integer>
- show lacp

Example:

```
set lacp -sysPriority 50 -ownerNode 2
<!--NeedCopy-->
```

To configure the LACP system priority by using the NetScaler GUI:

1. Navigate to System > Network > Interfaces and, in the Action list, select Set LACP.
2. Specify the system priority and the owner node (applicable only for a cluster setup).

Creating Link Aggregation Channels

For creating a link aggregation channel by using LACP, you need to enable LACP and specify the same LACP key on each interface that you want to be the part of the channel. For example, if you enable

LACP and set the LACP Key to 3 on interfaces 1/1 and 1/2, a link aggregation channel LA/3 is created and interfaces 1/1 and 1/2 are automatically bound to it.

Note:

- When enabling LACP on a network interface, you must specify the LACP Key.
- By default, LACP is disabled on all network interfaces.

To create an LACP channel by using the NetScaler command line:

At the command prompt, type:

- set interface <id> [-lacpMode <lacpMode>] [-lacpKey<positive_integer>] [-lacpPriority <positive_integer>] [-lacpTimeout (LONG | SHORT)]
- show interface [<id>]

To create an LACP channel by using the NetScaler GUI:

Navigate to System > Network > Interfaces, open the network interface, and set the parameters.

Modifying Link aggregation Channels

After you have created an LACP channel by specifying interfaces, you can modify properties of the channel.

To modify an LACP channel by using the NetScaler command line:

At the command prompt, type:

- set channel <id> [-ifnum <interfaceName> …] [-state (ENABLED | DISABLED)] [-speed <speed>] [-flowControl <flowControl>] [-haMonitor (ON | OFF)] [-ifAlias <string>] [-throughput <positive_integer>] [-tagall (ON | OFF)] [-bandwidthHigh <positive_integer> [-bandwidthNormal <positive_integer>]]
- show channel

Example:

```
> set channel LA/3 -state ENABLED -speed 10000
 Done
<!--NeedCopy-->
```

To modify an LACP channel by using the NetScaler GUI:

Navigate to System > Network > Channels, and modify an existing LACP channel.

Removing a Link Aggregation Channel

To remove a link aggregation channel that was created by using LACP, you need to disable LACP on all the interfaces that are part of the channel.

To remove an LACP channel by using the NetScaler command line:

At the command prompt, type:

- set interface <id> -lacpMode Disable
- show interface [<id>]

To remove an LACP channel by using the NetScaler GUI:

Navigate to System > Network > Interfaces, open the network interface, and clear the Enable LACP option.

Link Redundancy using LACP channels

Link Redundancy using LACP channels enables the NetScaler appliance to divide an LACP channel into logical subchannels, with one subchannel active and the others in standby mode. If the active subchannel fails to meet a minimum threshold of throughput, one of the standby subchannels becomes active and takes over.

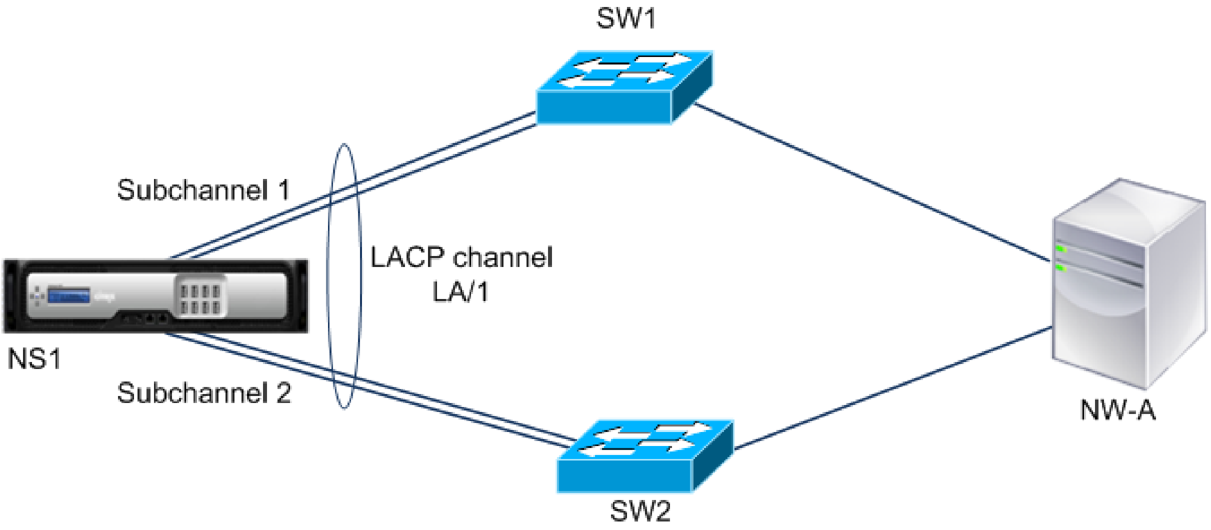
A subchannel is created from links that are part of the LACP channel and are connected to a particular device. For example, for an LACP channel with four interfaces on a NetScaler appliance, with two of the interfaces connected to device A and the other two connected to device B, the appliance creates two logical subchannels, one subchannel with two links to device A, and another subchannel with two links to device B.

To configure link redundancy for an LACP channel, set the lrMinThroughput parameter, which specifies the minimum throughput threshold (in Mbps) to be met by the active subchannel. Setting this parameter automatically creates the subchannels. When the maximum supported throughput of the active channel falls below the lrMinThroughput value, link failover occurs and a standby subchannel becomes active.

If you unset the lrMinThroughput parameter of an LACP channel, or set the value to zero, link redundancy for that channel is disabled, which is the default setting.

Example

Consider an example of link redundancy configured between NetScaler appliance NS1 and switches SW1 and SW2.



NS1 is connected to network device NW-A through SW1 and SW2.
On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000Mbps.
When the lrMinThroughput parameter is set to some value (say 2000), NS1 creates two logical subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2).

NS1 applies an algorithm to make one subchannel (say subchannel 1) active and put the other on standby. NS1 and network device NW-A are accessible to each other through only the active subchannel.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over.

Link Redundancy using LACP channels in a High Availability setup

In a high availability (HA) configuration, if you want to configure throughput (throughput parameter) based HA failover and link redundancy (lrMinThroughput parameter) on an LACP channel, you must set the throughput parameter to a value less than or equal to that of the lrMinThroughput parameter.

The maximum supported throughput of an LACP channel is calculated as the maximum supported throughput of the active subchannel.

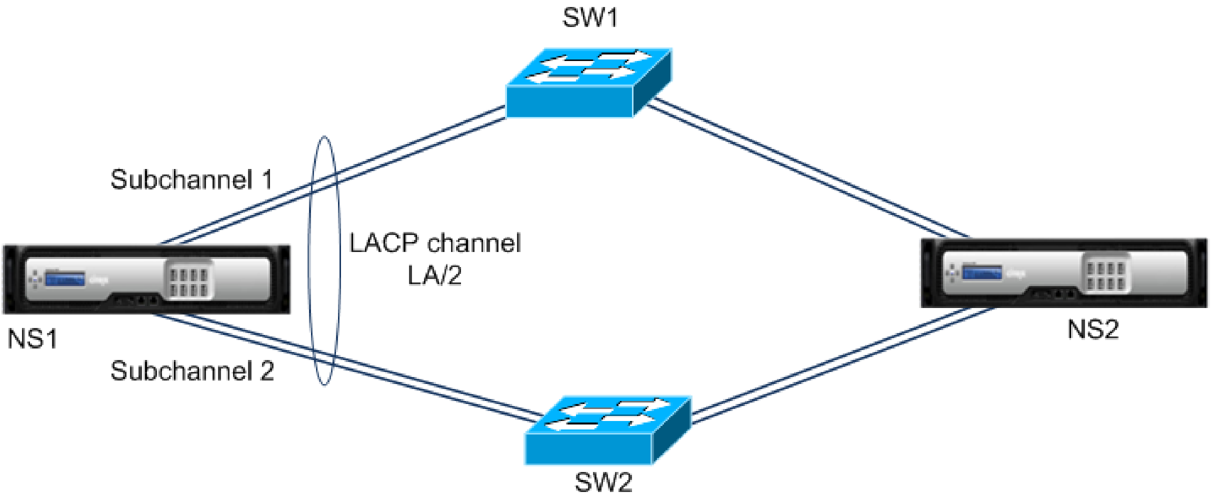
If the throughput parameter value is equal to or less than the lrminthroughput parameter value, HA failover occurs when both of the following conditions exist at the same time:

- None of the subchannels' maximum supported throughput meet the lrMinThroughput parameter value.

- The maximum supported throughput of the LACP channel does not meet the throughput parameter value

Consider an example of an HA setup that has NetScaler appliances NS1 and NS2, with switches SW1 and SW2. NS1 is connected to NS2 through SW1 and SW2.

On NS1, LACP channel LA/1 is created from interfaces 1/1, 1/2, 1/3, and 1/4. Interfaces 1/1 and 1/2 of NS1 are connected to SW1, and interfaces 1/3 and 1/4 are connected to SW2. Each of the four links supports a maximum throughput of 1000 Mbps.



Following are the LACP-parameter settings in this example:

Parameter	Value
Throughput	2000
lrminthroughput	2000

NS1 forms two subchannels from LA/1, one subchannel (say subchannel 1) using interfaces 1/1 and 1/2 (connected to SW1), and the other subchannel (subchannel 2) using interfaces 1/3 and 1/4 (connected to SW2). Each of the two subchannels supports a maximum throughput of 2000 Mbps. Applying an algorithm, NS1 makes one subchannel (say subchannel 1) active and the other standby.

Say subchannel 1 is active, and its maximum supported throughput falls below the lrMinThroughput value (for example, one of its links fails, and maximum supported throughput falls to 1000 Mbps). Subchannel 2 becomes active and takes over. HA failover does not occur, because the maximum supported throughput of the LACP channel is not less than the throughput parameter value:

Maximum supported throughput of the LACP channel = Maximum supported throughput of the active channel = Maximum supported throughput of subchannel 2 = 2000 Mbps

If subchannel 2's maximum supported throughput also falls below the lrminthroughput value (for example, one of its links fails, and the maximum supported throughput falls to 1000 Mbps), HA failover occurs, because the maximum supported throughput of the LACP channel is then less than the throughput parameter value:

Configure Link Redundancy using LACP channels

To configure link redundancy for a LACP channel by using the NetScaler command line:

At the command prompt, type the following commands to configure the channel and verify the configuration:

- **set channel** <id> -lrMinThroughput <positive_integer>
- **show channel**

Example:

```
> set channel la/1 - lrMinThroughput  2000
Done
> set channel la/2 - throughput 2000 - lrMinThroughput  2000
Done
<!--NeedCopy-->
```

To configure link redundancy for a LACP channel by using NetScaler GUI

1. Navigate to System > Network > Channels.
2. In the details pane, select an LACP channel for which you want to configure link redundancy, and then click Edit.
3. In the Configure LACP channel dialog box, set the lrMinThroughput parameter.
4. Click Close.

Redundant Interface Set

September 21, 2020

A redundant interface set is a set of interfaces where one of the interfaces is active and the remaining ones are standby. If the active interface fails, one of the standby interfaces takes over and becomes active.

The following are the main benefits of using redundant interface sets:

- A redundant interface set ensures connection reliability between the NetScaler appliance and a peer device by providing back up links between them.

- Unlike link redundancy using LACP, no configuration is required on the peer device for a redundant interface set. To the peer device, redundant interface set appear as individual interfaces and not as a set or collection.
- In an high availability configuration (HA), redundant interface sets can minimize the number the HA failovers.

> **Note**
>
> Redundant Interface Set was formerly known as 'NIC bundling' when first introduced in 10.5 release.

How Redundant Interface Set Works

For a redundant interface set, the NetScaler appliance derives a MAC address on the basis of an internal algorithm and assigns it to the redundant interface set. This MAC address is shared by all the member interfaces and is used only by the active interface at a time. The active interface broadcasts GARP messages, which contains the MAC address assigned to the redundant interface set and not the interface's own physical MAC address. When the current active interface fails and is taken over by another interface, the new active interface sends GARP messages. The peer device updates its forwarding table with the new active interface information. The standby interfaces do not send any GARP messages. The standby interfaces do not send any packets and they drop any packets they receive.

In a redundant interface set, selection of the member interface as active is based on either of the following factors:

- **Redundant interface priority**. This is a parameter of an interface and it defines the priority of the interface in a redundant interface set for the active member selection. This parameter specifies a positive integer. Lower the value higher the priority of active member selection. The member interface with the highest priority (lowest value) is selected as the active interface of the redundant interface set.
- **Binding order of the member interfaces**. If all the member interfaces have the same redundant interface priority, the member interface that was bound first to the redundant interface set is selected as the active interface of the redundant interface set.

In a redundant interface set, active interface selection is triggered in one of the following events:

- When the current active interface fails or you disable it.
- When you set the priority of a standby interface to a value lower than that of the current active interface. The standby interface takes over as the active interface.
- When you bind an interface whose priority is lower than that of the current active interface. The newly bound interface takes over as the active interface.

Points to Consider for Configuring Redundant Interface Sets

Consider the following points before you configure a redundant interface set:

- In a standalone appliance or an appliance in a high availability setup, a link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.
- In a high availability configuration, redundant interface set configurations do not propagate or synchronize to the secondary node.
- You can configure a maximum of four redundant interface sets on a NetScaler appliance.
- You can bind a maximum of 16 interfaces to a redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to another redundant interface set.
- Member interfaces of a redundant interface set cannot be bound to a link aggregate (LA) channel.
- LA channels cannot be bound to a redundant interface set.
- Redundant interface sets cannot be bound to an LA channel.
- In a cluster setup:
  - Redundant interface sets cannot be bound to a cluster link aggregation.
  - A link redundant set is specified in N/LR/X notation (for example, 1/LR/3). Where:
    N is the ID of the cluster node on which the redundant interface set is to be created.
    X is a link-redundant set identifier on a cluster node. X can range from 1-4.
  - A cluster link aggregation cannot be bound to a redundant interface set.
  - A redundant interface set can include only the interfaces of the node to which the redundant interface set belongs.
  - An Existing elink redundancy set configuration on a standalone appliance automatically changes to cluster notation (N/LR/X) after the appliance is added to a cluster setup.

Configuration Steps

Configuring redundant interface set on a NetScaler appliance consists of the following tasks:

- **Create a redundant interface set**. Use the channel command operation for creating a redundant interface set.

  In a standalone appliance or an appliance in a high availability setup, a link redundant set is specified in LR/X notation, where X can range from 1 to 4. For example, LR/1.

  In a cluster setup, a link redundant set is specified in N/LR/X (for example, 1/LR/3), where: N is the ID of the cluster node on which the redundant interface set is to be created, X is link redundant set identifier on a cluster node. X can range from 1-4.

- **Bind interfaces to the redundant interface set**. Associate the desired interfaces with the redundant interface set. An interface cannot be a part of multiple redundant interface sets.

- **(Optional) Set a redundant interface priority on the member interface**. Use the interface command operation for setting the redundant interface priority on a desired member interface of a redundant interface set.

To create a redundant interface set by using the NetScaler command line:

At the command prompt:

- add channel <ID>
- show channel <ID>

To bind interfaces to a redundant interface set by using the NetScaler command line:

At the command prompt:

- bind channel <ID> <ifnum>
- show channel <ID>

To set a redundant interface priority of an interface by using the NetScaler command line:

At the command prompt:

- set interface <ID> -lrsetpriority <positive_integer>
- show interface <ID>

Sample configuration 1:

In the following example, redundant interface set LR/1 is created, and interfaces 1/1, 1/2, 1/3, and 1/4 are bound to LR/1. The redundant interface priority is set to a default value of 1024 for all these member interfaces. Output of the show channel command displays that the interface 1/1 is the current active interface for the redundant interface set lr/1.

```
> add channel lr/1
 Done
> bind channel lr/1 1/1 1/2 1/3 1/4
 Done
> show channel
1)    Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
```

```
                1/1: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Active Member
                1/2: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
Done
<!--NeedCopy-->
```

Sample configuration 2:

In the following example, redundant interface priority of the member interface 1/4 is set to 100, which is lower than the set redundant interface priority of all the other member interfaces of LR/1.

Output of the show channel command displays that the interface 1/4 is the current active interface for the redundant interface set LR/1.

```
> set interface 1/4 -lrsetPriority 100
Done
> show channel
1)      Interface LR/1 (Link Redundant) #23
        flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
        MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
        Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
                 throughput 0
        Actual: throughput 1000
        LLDP Mode: NONE,
        RX: Pkts(1) Bytes(52) Errs(0) Drops(1) Stalls(0)
        TX: Pkts(2) Bytes(84) Errs(0) Drops(4) Stalls(0)
        NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
        Bandwidth thresholds are not set.
                1/1: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
                1/2: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
                1/3: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Inactive Member
                1/4: UTP-1000-FULL-OFF         UP  0h14m06s    LR
                   Active Member
Done
<!--NeedCopy-->
```

Sample configuration 3:

Consider a cluster setup of four nodes N1, N2, N3, and N4. In this example, redundant interface set 1/LR/3 is created on node N1, and interfaces 1/1/1, 1/1/2, and 1/1/3 are bound to it. The redundant interface priority is set to a default value of 1024 for all these member interfaces. Output of the show channel command indicates that interface 1/1/1 is the current active interface for redundant interface set 1/LR/3.

```
    > add channel 1/LR/3

    Done
    > bind channel 1/LR/3 1/1/1 1/1/2 1/1/3

    Done
    > show channel
    1)    Interface 1/LR/3 (Link Redundant) #14
            flags=0x100c020 <ENABLED, UP, LINKREDUNDANT, UP, HAMON, 802.1q>
            MTU=1500, native vlan=1, MAC=36:97:a2:b7:6b:a9, uptime 0h00m00s
            Requested: media NONE, speed AUTO, duplex NONE, fctl OFF,
            throughput 0
            Actual: throughput 1000
            LLDP Mode: NONE,
            RX: Pkts(66) Bytes(4406) Errs(0) Drops(82) Stalls(0)
            TX: Pkts(55) Bytes(2626) Errs(0) Drops(145) Stalls(0)
            NIC: InDisc(0) OutDisc(0) Fctls(0) Stalls(0) Hangs(0) Muted(0)
            Bandwidth thresholds are not set.

            1/1/1: UTP-1000-FULL-OFF UP  0h14m06s LR Active Member
            1/1/2: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member
            1/1/3: UTP-1000-FULL-OFF UP  0h14m06s LR Inactive Member

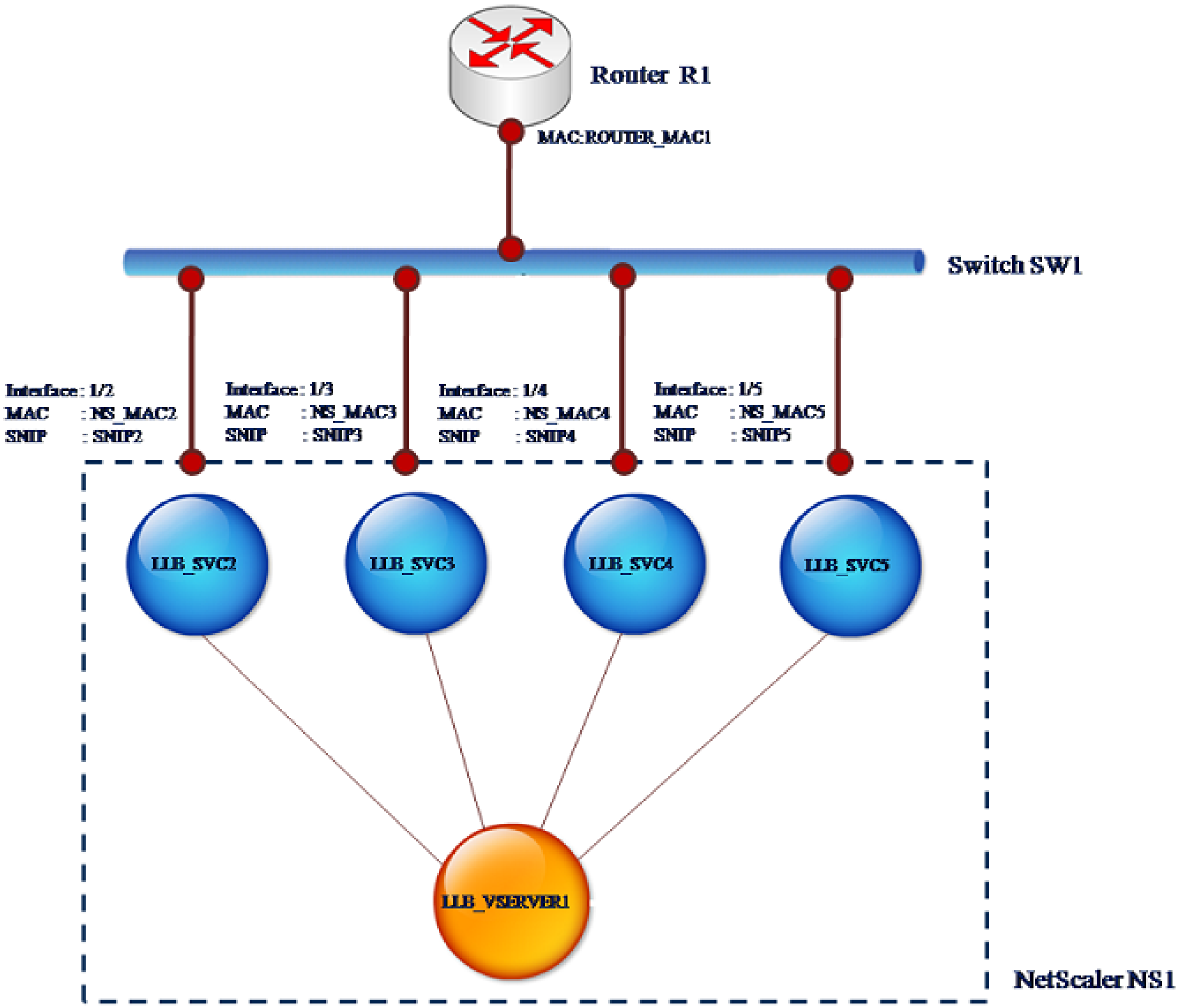
    Done
<!--NeedCopy-->
```

Binding an SNIP address to an Interface

September 21, 2020

You can now bind a NetScaler owned SNIP address to an interface without using Layer 3 VLANs. Any packets related to the SNIP address will go only through the bound interface.

This feature can be useful in a scenario where the upstream switch does not support Link Aggregation channels and you want the NetScaler appliance to load balance traffic, originated from a server, across the four links to the upstream switch as shown in the following illustration.


Router R1
MAC:ROUTER_MAC1
Switch SW1
Interface: 1/2
MAC : NS_MAC2
SNIP : SNIP2
Interface: 1/3
MAC : NS_MAC3
SNIP : SNIP3
Interface: 1/4
MAC : NS_MAC4
SNIP : SNIP4
Interface: 1/5
MAC : NS_MAC5
SNIP : SNIP5
LLB_SVC2
LLB_SVC3
LLB_SVC4
LLB_SVC5
LLB_VSERVER1
NetScaler NS1

The following tables describe the example settings for the scenario:

Entity	Name	Value
SNIP addresses on NS1	SNIP2 (for reference purpose only)	10.10.10.2
	SNIP3 (for reference purpose only)	10.10.10.3
	SNIP4 (for reference purpose only)	10.10.10.4
	SNIP5 (for reference purpose only)	10.10.10.5

Entity	Name	Value
LLB virtual server on NS1	LLB_VSERVER1	-
Transparent monitor on NS1	TRANS_MON	-
LLB services on NS1	LLB_SVC2	10.10.10.240
	LLB_SVC3	10.10.10.120
	LLB_SVC4	10.10.10.60
	LLB_SVC5	10.10.10.30
MAC address of interface 1/2 on NS1	NS_MAC_2 (for reference purpose only)	00:e0:ed:0f:bc:e0
MAC address of interface 1/3 on NS1	NS_MAC_3 (for reference purpose only)	00:e0:ed:0f:bc:df
MAC address of interface 1/4 on NS1	NS_MAC_4 (for reference purpose only)	00:e0:ed:0f:bc:de
MAC address of interface 1/5 on NS1	NS_MAC_5 (for reference purpose only)	00:e0:ed:1c:89:53
IP address of Router R1	Router_IP (for reference purpose only)	10.10.10.1
MAC address of interface of R1	ROUTER_MAC1 (for reference purpose only)	00:21:a1:2d:db:cc

To configure the example settings:

1. Add four different SNIPs in different subnet ranges. This is for ARP to be resolved on four different links. For more information on creating a SNIP address, see Configuring Subnet IP Addresses (SNIPs).

   NetScaler command line example:

   ```
   > add ns ip 10.10.10.2 255.255.255.0 -type SNIP
    Done
   > add ns ip 10.10.10.3 255.255.255.128 - type SNIP
   Done
   > add ns ip 10.10.10.4 255.255.255.192 - type SNIP
   Done
   > add ns ip 10.10.10.5 255.255.255.224 - type SNIP
   Done
   <!--NeedCopy-->
   ```

2. Add four different dummy services in the added SNIP subnets. This is to ensure that the traffic is sent out with source IP as one of the four configured SNIPs. For more information on creating a service, see Set up basic load balancing.

   NetScaler command line example:

   ```
   > add service LLB_SVC2 10.10.10.240 any *
    Done
   > add service LLB_SVC3 10.10.10.120 any *
    Done
   > add service LLB_SVC4 10.10.10.60 any *
    Done
   > add service LLB_SVC5 10.10.10.30 any *
    Done
   <!--NeedCopy-->
   ```

3. Add a transparent ping monitor for monitoring the gateway. Bind the monitor to each of the configured dummy services. This is to make the state of the services as UP. For more information on creating a transparent monitor, see Configure monitors in a load balancing setup.

   NetScaler command line example:

   ```
   > add monitor TRANS_MON ping -destIP 10.10.10.1 -transparent YES
    Done
   > bind monitor TRANS_MON LLB_SVC2
    Done
   > bind monitor TRANS_MON LLB_SVC3
    Done
   > bind monitor TRANS_MON LLB_SVC4
    Done
   > bind monitor TRANS_MON LLB_SVC5
    Done
   <!--NeedCopy-->
   ```

4. Add a link load balancing (LLB) virtual server and bind the dummy services to it. For more information on creating an LLB virtual server, see Configuring a Basic LLB Setup.

   NetScaler command line example:

   ```
   > add lb vserver LLB_VSERVER1 any
    Done
   > set lb vserver LLB_VSERVER1 -lbmethod ROUNDROBIN
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
    Done
   > bind lb vserver LLB_VSERVER1 LLB_SVC2
   ```

```
Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
Done
> bind lb vserver LLB_VSERVER1 LLB_SVC2
Done
<!--NeedCopy-->
```

5. Add the LLB virtual server as the default LLB route. For more information on creating an LLB route see Configuring a Basic LLB Setup.

   NetScaler command line example:

   ```
   > add lb route 0.0.0.0 0.0.0.0  LLB_VSERVER1
   Done
   <!--NeedCopy-->
   ```

6. Add an ARP entry for each of the dummy services with the MAC address of the gateway. This way the gateway is reachable through these dummy services. For more information on adding an ARP entry, see Configuring Static ARP.

   NetScaler command line example:

   ```
   > add arp -ipaddress 10.10.10.240 -mac 00:21:a1:2d:db:cc -ifnum
     1/2
   Done
   > add arp -ipaddress 10.10.10.120 -mac 00:21:a1:2d:db:cc -ifnum
     1/3
   Done
   > add arp -ipaddress 10.10.10.60 -mac 00:21:a1:2d:db:cc -ifnum 1/4
   Done
   > add arp -ipaddress 10.10.10.30 -mac 00:21:a1:2d:db:cc -ifnum 1/5
   Done
   <!--NeedCopy-->
   ```

7. Bind a specific interface to an SNIP by adding an ARP entry for each of these SNIPs. This is to ensure that the response traffic will reach the same interface through which the request went out. For more information on adding an ARP entry, see Configuring Static ARP.

   NetScaler command line example:

   ```
   > add arp -ipAddress 10.10.10.2 -mac 00:e0:ed:0f:bc:e0 -ifnum 1/2
   Done
   > add arp -ipAddress 10.10.10.3 -mac 00:e0:ed:0f:bc:df -ifnum 1/3
   Done
   > add arp -ipAddress 10.10.10.4 -mac 00:e0:ed:0f:bc:de -ifnum 1/4
   Done
   ```

```
> add arp -ipAddress 10.10.10.5 -mac 00:e0:ed:1c:89:53 -ifnum 1/5
 Done
<!--NeedCopy-->
```

Monitor the Bridge Table and Changing the Aging time

September 21, 2020

NetScaler appliance bridges frames on the basis of bridge table lookup of the destination MAC address and the VLAN ID. However, the appliance performs forwarding only when Layer 2 mode is enabled.

The bridge table is dynamically generated, but you can display it, modify the aging time for the bridge table, and view bridging statistics. All the MAC entries in the bridge table are updated with the aging time.

To set the aging time of bridge table entries by using the NetScaler command line:

At the command prompt, type:

- **set l2param** -**bridgeagetimeout** <positive_integer>
- **show l2param**

Example:

```
> set l2param -bridgeagetimeout 90
 Done
<!--NeedCopy-->
```

To view the statistics of a bridge table by using the NetScaler command line:

At the command prompt, type:

- **stat bridge**

To set the aging time of bridge table entries by using the NetScaler GUI:

Navigate to **System** > **Network**. On the **Network** page, in the **Settings** section, click **Configure Layer2 Parameters** and set the **Timeout Value For The Bridge Table Entries (seconds)** parameter.

To view the statistics of a bridge table by using the NetScaler GUI:

Navigate to **System** > **Network** > **Bridge Table**, select the MAC address, and click **Statistics**.

NetScaler Appliances in Active-Active Mode Using VRRP

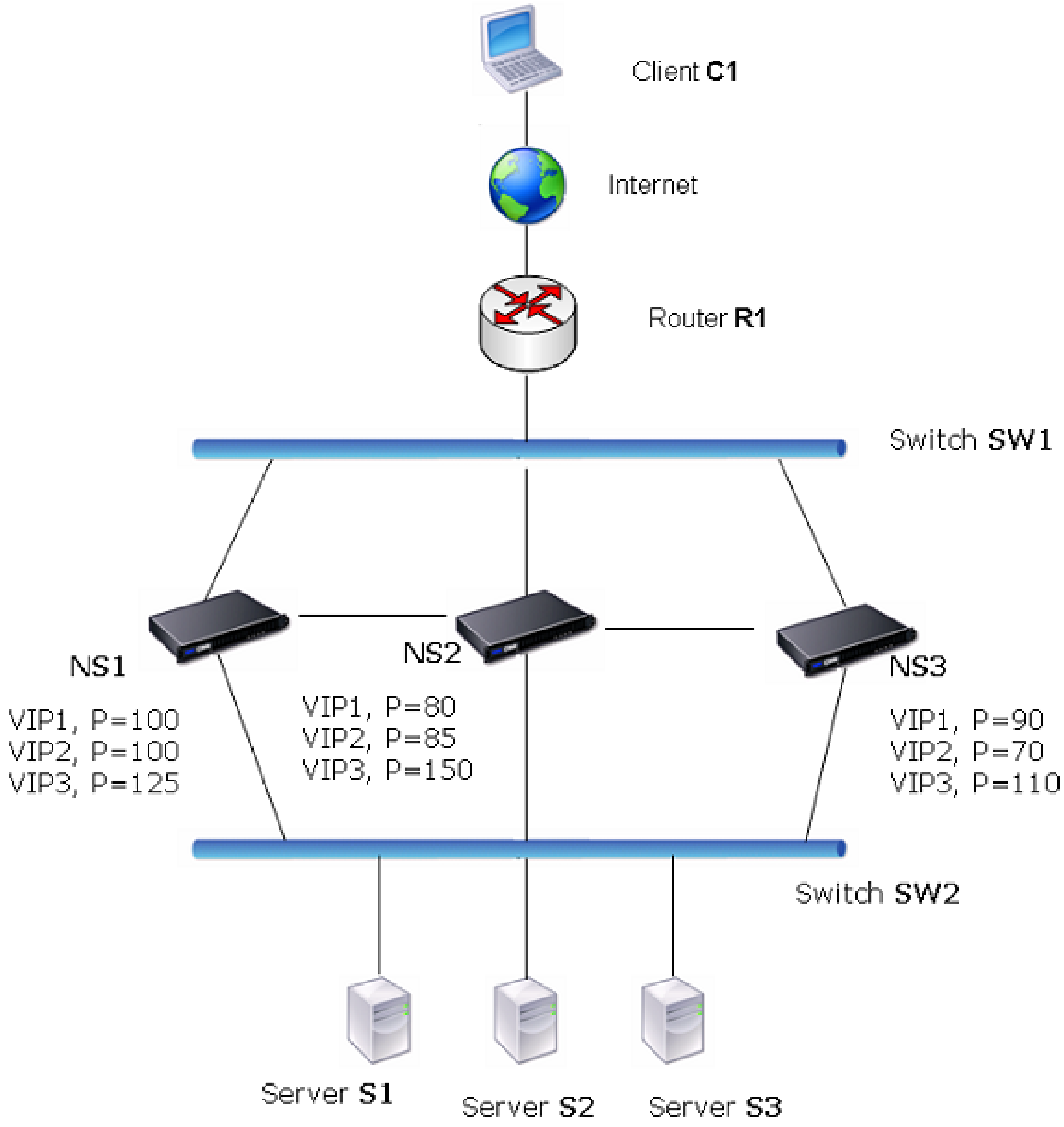
September 21, 2020

An active-active deployment, in addition to preventing downtime, makes efficient use of all the NetScaler appliances in the deployment. In active-active deployment mode, the same VIPs are configured on all NetScaler appliances in the configuration, but with different priorities, so that a given VIP can be active on only one appliance at a time.

The active VIP is called the master VIP, and the corresponding VIPs on the other NetScaler appliances are called the backup VIPs. If a master VIP fails, the backup VIP with the highest priority takes over and becomes the master VIP. All the NetScaler appliances in an active-active deployment use the Virtual Router Redundancy Protocol (VRRP) protocol to advertise their VIPs and the corresponding priorities at regular intervals.

NetScaler appliances in active-active mode can be configured so that no NetScaler is idle. In this configuration, different sets of VIPs are active on each NetScaler. For example, in the following diagram, VIP1, VIP2, VIP3, and VIP4 are configured on appliances NS1, NS2, and NS3. Because of their priorities, VIP1 and VIP 2 are active on NS1, VIP3 is active on NS2 and VIP 4 is active on NS3. If, for example, NS1 fails, VIP1 on NS3 and VIP2 on NS2 become active.

Figure 1. An Active-Active Configuration


Client C1
Internet
Router R1
Switch SW1
NS1
VIP1, P=100
VIP2, P=100
VIP3, P=125
NS2
VIP1, P=80
VIP2, P=85
VIP3, P=150
NS3
VIP1, P=90
VIP2, P=70
VIP3, P=110
Switch SW2
Server S1
Server S2
Server S3

The NetScaler appliances in the above diagram process traffic as follows:

1. Client C1 sends a request to VIP1. The request reaches R1.
2. R1 does not have an ARP entry for VIP1, so it broadcasts an ARP request for VIP1.
3. VIP1 is active in NS1, so NS1 replies with a source MAC address as the virtual MAC (for example VMAC1) associated with VIP1, and VIP1 as the source IP address.
4. SW1 learns the port for VIP1 from the ARP reply and updates its bridge table.
5. R1 updates the ARP entry with VMAC1 and VIP1.

6. R1 forwards the packet to the VIP1 on NS1.
7. NS1's load balancing algorithm selects server S2, and NS1 opens a connection between one of its SNIP addresses and S2.
8. S2 replies to the SNIP on the NetScaler.
9. NS1 sends S2's reply to the client. In the reply, NS1 inserts MAC address of the physical interface as the source MAC address and VIP1 as the source IP address.
10. Should NS1 fail, the NetScaler appliances use the VRRP protocol to select the VIP1 with the highest priority. In this case, VIP1 on NS3 becomes active, and the following two steps update the active-active configuration.
11. NS3 broadcasts a GARP message for VIP1. In the message, VMAC1 is the source MAC address and VIP1 is the source IP address.
12. SW1 learns the new port for VMAC1 from the GARP broadcast and updates its bridge table to send subsequent client requests for VIP1 to NS3. R1 updates its ARP table.

The priority of a VIP can be modified by health tracking. If you enable health tracking, you should make sure that preemption is also enabled, so that a VIP whose priority is lowered can be preempted by another VIP.

In some situations, traffic might reach a backup VIP. To avoid dropping such traffic, you can enable sharing, on a per-node basis, as you create an active-active configuration. Or you can enable the global send to master option. On a node on which sharing is enabled, it takes precedence over send to master.

Health Tracking

Base priority (BP-range 1-255) ordinarily determines which VIP is the master VIP, but effective priority (EP) can also affect the determination.

For example, if a VIP on NS1 has a priority of 101 and same VIP on NS2 has a priority of 99, the VIP on NS1 is active. However, if two vservers are using the VIP on NS1 and one of them goes DOWN, health tracking can reduce the EP of VIP on NS1. VRRP then makes the VIP on NS2 the active VIP.

Following are the health tracking options for modifying EP:

- **NONE.** No tracking. EP = BP
- **ALL.** If all virtual servers are UP, then EP = BP. Otherwise, EP = 0.
- **ONE.** If at least one virtual server is UP, then EP = BP. Otherwise, EP = 0.
- **PROGRESSIVE.** If ALL virtual servers are UP, then EP = BP. If ALL virtual servers are DOWN then EP = 0. Otherwise EP = BP (1 - K/N), where N is the total number of virtual servers associated with the VIP and k is the number of virtual servers that are down.

Note: If you specify a value other than NONE, preemption should be enabled, so that the backup VIP with the highest priority becomes active if the priority of the master VIP is downgraded.

Preemption

Preemption of an active VIP by another VIP that attains a higher priority is enabled by default, and normally should be enabled. In some cases, however, you may want to disable it. Preemption is a per-node setting for each VIP.

Preemption can occur in the following situations:

- An active VIP goes down and a VIP with a lower priority takes its place. If the VIP with the higher priority comes back online, it preempts the currently active VIP.
- Health tracking causes the priority of a backup VIP to become higher than that of the active VIP. The backup VIP then preempts the active VIP.

Sharing

In the event that traffic reaches a backup VIP, the traffic is dropped unless the sharing option is enabled on the backup VIP. This behavior is a per node setting for each VIP and is disabled by default.

In the figure **An Active-Active Configuration** VIP1 on NS1 is active and VIP1 VIPs on NS2 and NS3 are backups. Under certain circumstances, traffic may reach VIP1 on NS2. If Sharing is enabled on NS2, this traffic is processed instead of dropped.

Configuring Active-Active Mode

September 21, 2020

On each NetScaler appliance that you want to deploy in active-active mode, you must add a Virtual MAC and bind the virtual MAC to a VIP. The virtual MAC for a given VIP must be same on each appliance. For example, if VIP 10.102.29.5, is created on the appliances, a virtual router ID (VRID) must be created on each NetScaler and bound to VIP 10.102.29.5 on each NetScaler. When you bind a virtual MAC to a VIP, the NetScaler sends VRRP advertisements to each VLAN that is bound to that VIP. The virtual MAC can be shared by different VIPs configured on the same NetScaler.

Configuring IPv4 Active-Active Mode

Perform the following tasks on each of the NetScaler appliances to be included in the active-active configuration:

- **Add a virtual MAC address**. Add a virtual MAC address by adding a VRID. You can also specify a priority and enable or disable preemption and sharing on this VRID address.

- **Add a VIP address and associate the Virtual MAC's VRID**. Add a VIP address and set the VRID parameter to the newly created VRID. The attributes of the VRID (for example, priority and preemption) are bound to this VIP address.
  Note: The same VIP address must be added to all the other NetScaler appliances.

To add a virtual MAC address by using the NetScaler command line

At the command prompt, type:

- **add vrID** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED**|**DISABLED**)][-**sharing** (**ENABLED**|**DISABLED**)] [-**tracking** <tracking>]
- **show vrid**

To add a VIP address by using the NetScaler command line:

At the command prompt, type:

- **add ns ip** <IPv4Address> -type VIP -vrid <value>
- **show ns ip**

To configure a virtual MAC by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC**, on the **VMAC** tab, add a new Virtual MAC, or edit an existing Virtual MAC.
2. Set the following parameters:
   - Virtual Router ID
   - Priority
   - Tracking
   - Preemption
   - Sharing

To configure a VIP address and associate the VRID to it by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV4s** tab, add an IP address of type VIP.
2. While adding the IP address, select the virtual router ID from the **Virtual Router Id** drop down box.

Sample Configuration:

The following sample configuration is for deploying NetScaler appliances NS1 and NS2 in IPv4 active-active mode. VIP address 203.0.113.10 is configured on both NS1 and NS2, with a different priority value on each appliance. On each appliance, this VIP address is bound to a virtual MAC address. 203.0.113.10 is master on NS2, because its priority (200) on NS2 is higher than on NS1 (100).

```
Settings on NS1

> add vrid 10 - Priority 100 - Preemption Enabled - sharing Enabled

```

```
    Done

    > add ns ip 203.0.113.10 - type VIP - vrid 10

    Done

    Settings on NS2

    > add vrid 10 - Priority 200 - Preemption Enabled - sharing Enabled

    Done

    > add ns ip 203.0.113.10 - type VIP - vrid 10

    Done
<!--NeedCopy-->
```

Configuring IPv6 Active-Active Mode

Perform the following tasks on each of the NetScaler appliances to be included in the active-active configuration:

- **Add a Vitrtual MAC6 address**. Add a Vitrtual MAC6 address by adding a VRID6. You can also specify a priority and enable or disable preemption and sharing on this VRID6 address.
- **Add a VIP6 address**. Add a VIP6 address. Set the VRID6 parameter to the VRID6 of the newly created Virtual MAC6.The attributes of the Vitrtual MAC6 (for example, priority and preemption) are bound to this VIP6 address.
  Note: The same VIP6 address must be added to all the other NetScaler appliances.

To add a Vitrtual MAC6 address by using the NetScaler command line:

At the command prompt, type:

- **add vrID6** <id> [-**priority** <positive_integer>] [-**preemption** (**ENABLED** | **DISABLED**)] [-**sharing** (**ENABLED** | **DISABLED**)]
- **show vrID6**

To add a VIP6 address by using the NetScaler command line:

At the command prompt, type:

- **add ns ip6** <IPv6Address> -**type** VIP -**vrid** <value>
- **show ns ip6**

To configure a Vitrtual MAC6 by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC**, on the **VMAC6** tab, add a new Virtual MAC6, or edit an existing **VMAC6**.
2. Set the following parameters:
    - Virtual Router ID
    - Priority
    - Preemption
    - Sharing

To configure a VIP6 address and associate the VRID to it by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV6s** tab, add an IPv6 address of type VIP.
2. While adding the VIP6 address, select the VRID6 from the **Virtual Router Id** drop down box.

Sample Configuration:

The following sample configuration is for deploying NetScaler appliances NS1 and NS2 in IPv6 active-active mode. VIP6 address 2001:db8::5001 is configured on both NS1 and NS2, with a different priority value on each appliance. On each appliance, this VIP6 address is bound to a Vitrtual MAC6 address. 2001:db8::5001 is master on NS2, because it's priority (200) on NS2 is higher than on NS1 (100).

```
  Settings on NS1
  > add vrid6 10 - Priority 100 - Preemption Enable - sharing Enable

  Done
  > add ns ip6 2001:db8::5001 - type VIP - vrid6 10

  Done
  Settings on NS2
  > add vrid6 10 - Priority 200 - Preemption Enable - sharing Enable

  Done
  > add ns ip6 2001:db8::5001 - type VIP - vrid6 10

  Done
<!--NeedCopy-->
```

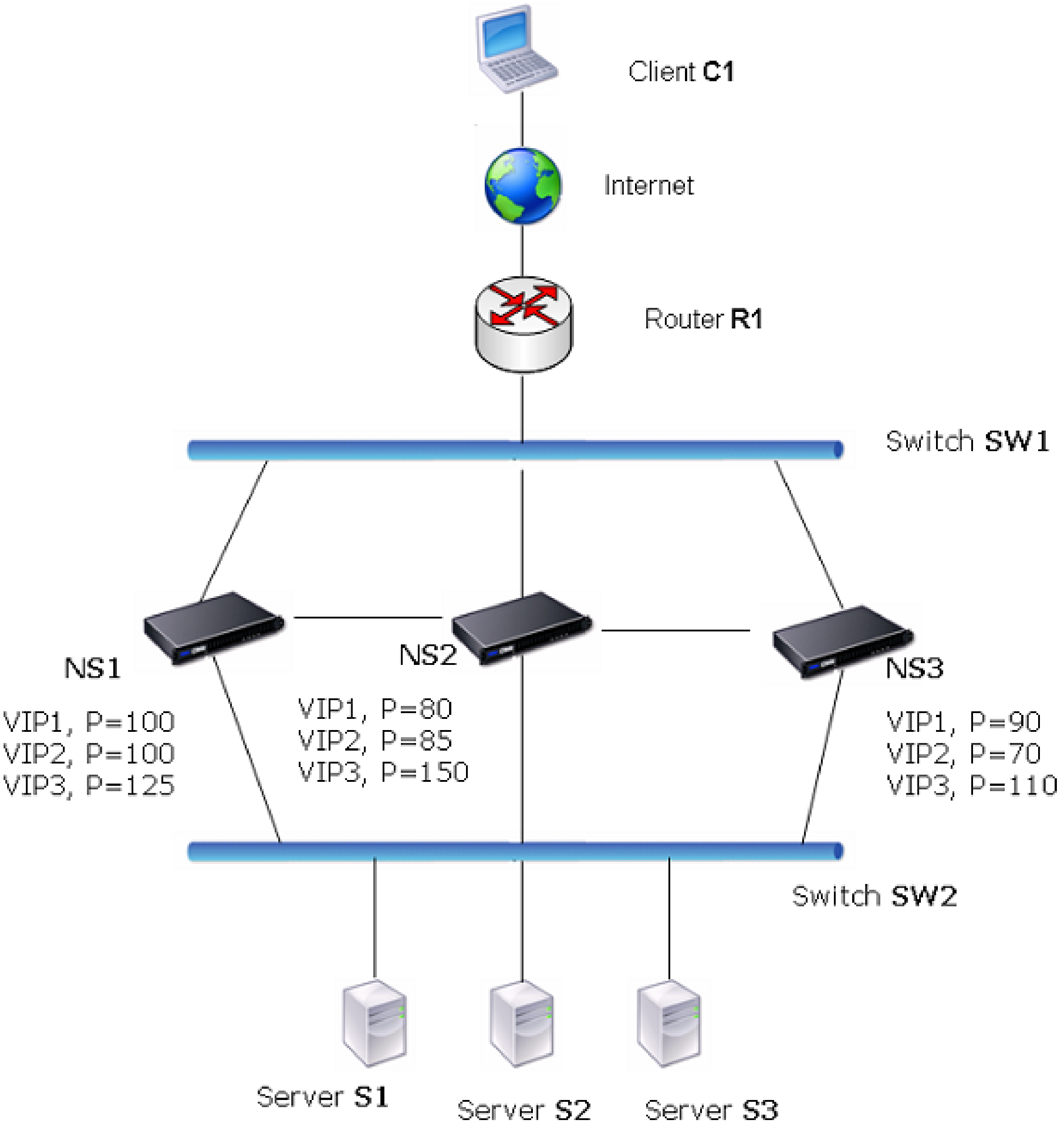
Configuring Send to Master

September 21, 2020

Usually, the traffic destined to a VIP reaches the NetScaler appliance on which the VIP is active, because an ARP request with the VIP and a virtual MAC on that appliance has reached the upstream router. But in some cases, such as static routes configured on the upstream router for the VIP subnet, or a topology that blocks this route, the traffic can reach a NetScaler appliance on which the VIP is in backup state. If you want this appliance to forward the data packets to the appliance on which the VIP is active, you need to enable the send to master option. This behavior is a per node setting and is disabled by default.

For example, in the following diagram, VIP1 is configured on NS1, NS2, and NS3 and is active on NS1. Under certain circumstances, traffic for VIP1 (active on NS1) may reach VIP1 on NS3. When the send to master option is enabled on NS3, NS3 forwards the traffic to NS1 through NS2 by using route entries for NS1.

Figure 1. An Active-Active Configuration with Send to Master Option Enabled


Client C1
Internet
Router R1
Switch SW1
NS1
VIP1, P=100
VIP2, P=100
VIP3, P=125
NS2
VIP1, P=80
VIP2, P=85
VIP3, P=150
NS3
VIP1, P=90
VIP2, P=70
VIP3, P=110
Switch SW2
Server S1
Server S2
Server S3

To enable send to master by using the NetScaler command line:

At the command prompt, type:

```
set vrIDParam -sendToMaster (ENABLED       DISABLED)
```

Example:

```
> set vrIDParam -sendToMaster ENABLED
 Done
<!--NeedCopy-->
```

To enable send to master by using the NetScaler GUI:

1. Navigate to **System** > **Network**, in the **Settings** group, click **Virtual Router Parameters**.
2. Select the **Send to Master** option.

Configuring VRRP Communication Intervals

September 21, 2020

In an active-active deployment, all NetScaler nodes use the Virtual Router Redundancy Protocol (VRRP) to advertise their master VIP addresses and the corresponding priorities in VRRP advertisement packets (hello messages) at regular intervals.

VRRP uses the following communication intervals:

- **Hello Interval**. Interval between the VRRP hello messages that a node of a master VIP address sends to its peer nodes.
- **Dead Interval**. Time after which a node of a backup VIP address considers the state of the master VIP address as DOWN if VRRP hello messages are not received from the node of the master VIP address. After the dead interval, the backup VIP address takes over and becomes the master VIP address.

You can change these intervals to a desired value. Both of these communication intervals are per node setting for all VIP addresses in that node.

To configure VRRP communication intervals by using the NetScaler command line:

At the command prompt, type:

- **set vrIDParam** [-**helloInterval** <msecs>] [-**deadInterval** <secs>]
- **sh vrIDParam**

Example:

```
> set vrIDParam -helloInterval 500 -deadInterval 2
 Done
<!--NeedCopy-->
```

To configure VRRP communication intervals by using the NetScaler GUI:

1. Navigate to **System** > **Network**, in the **Settings** group, click **Virtual Router Parameters**.
2. In **Configure Virtual Router Parameter**, set the **Hello Interval** and **Dead Interval** parameters.

3. Click **OK**.

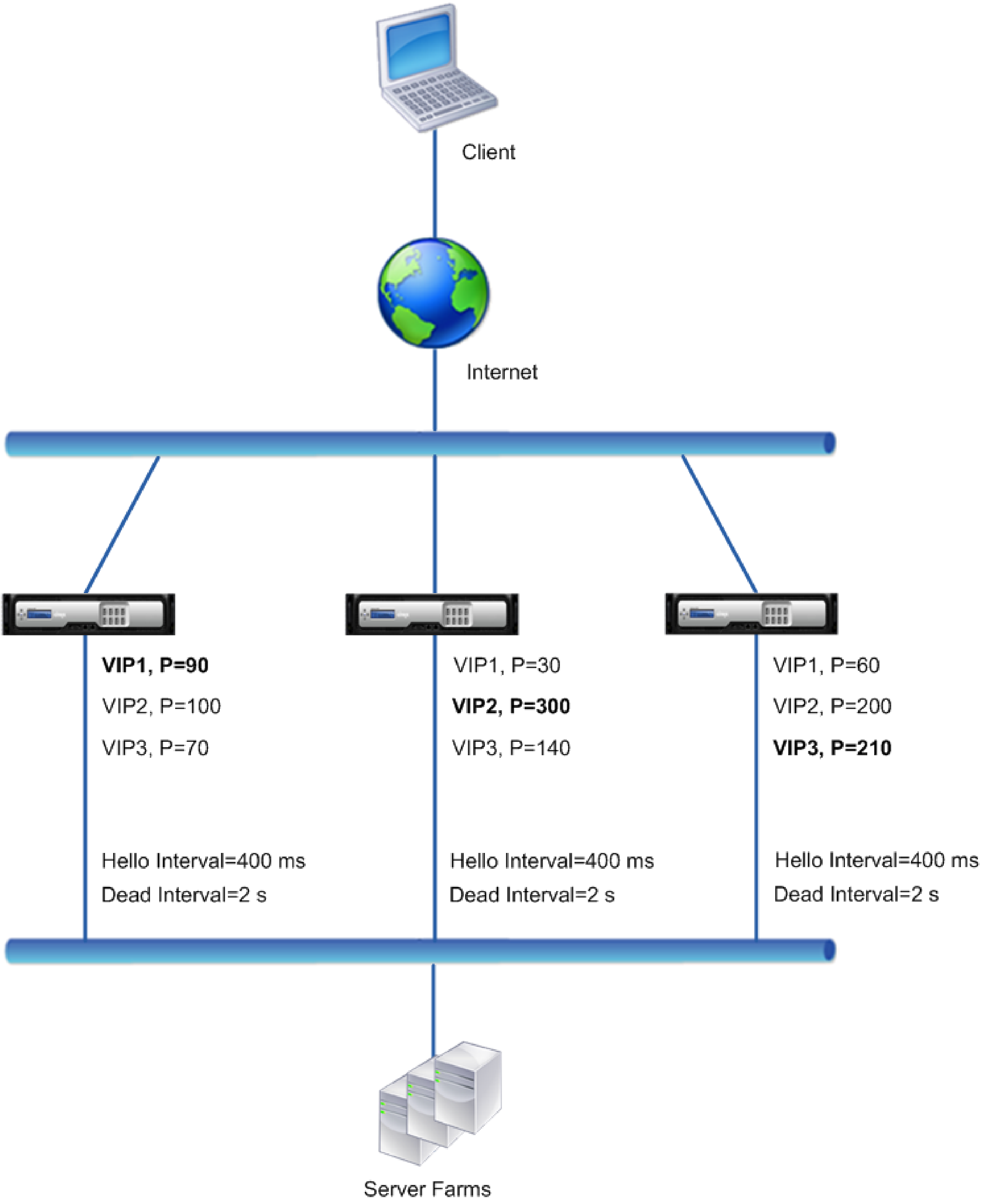
Example 1: Nodes with the Same VRRP Dead Intervals

Consider an active-active deployment consisting of NetScaler appliances NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3 are configured on each of these appliances. Because of their priorities, VIP1 is active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

As shown in the table below, the dead interval is set to the same value (2 seconds) on all the three nodes. The VRRP communication intervals (hello interval and dead interval) of a node apply to all the VRIDs configured on the node, and in turn, apply to all VIP addresses associated with the VRIDs on the node.

On each node, the VIP addresses that are active (master) on that node use the hello interval, and the dead interval is used by the VIP addresses that are inactive (backup) on that node. Preemption is disabled for the VIP addresses in all the three nodes.

The following table lists the settings used in this example: VRRP interval example 1 settings.


Client
Internet
VIP1, P=90
VIP2, P=100
VIP3, P=70
Hello Interval=400 ms
Dead Interval=2 s
VIP1, P=30
VIP2, P=300
VIP3, P=140
Hello Interval=400 ms
Dead Interval=2 s
VIP1, P=60
VIP2, P=200
VIP3, P=210
Hello Interval=400 ms
Dead Interval=2 s
Server Farms

The execution flow is as follows:

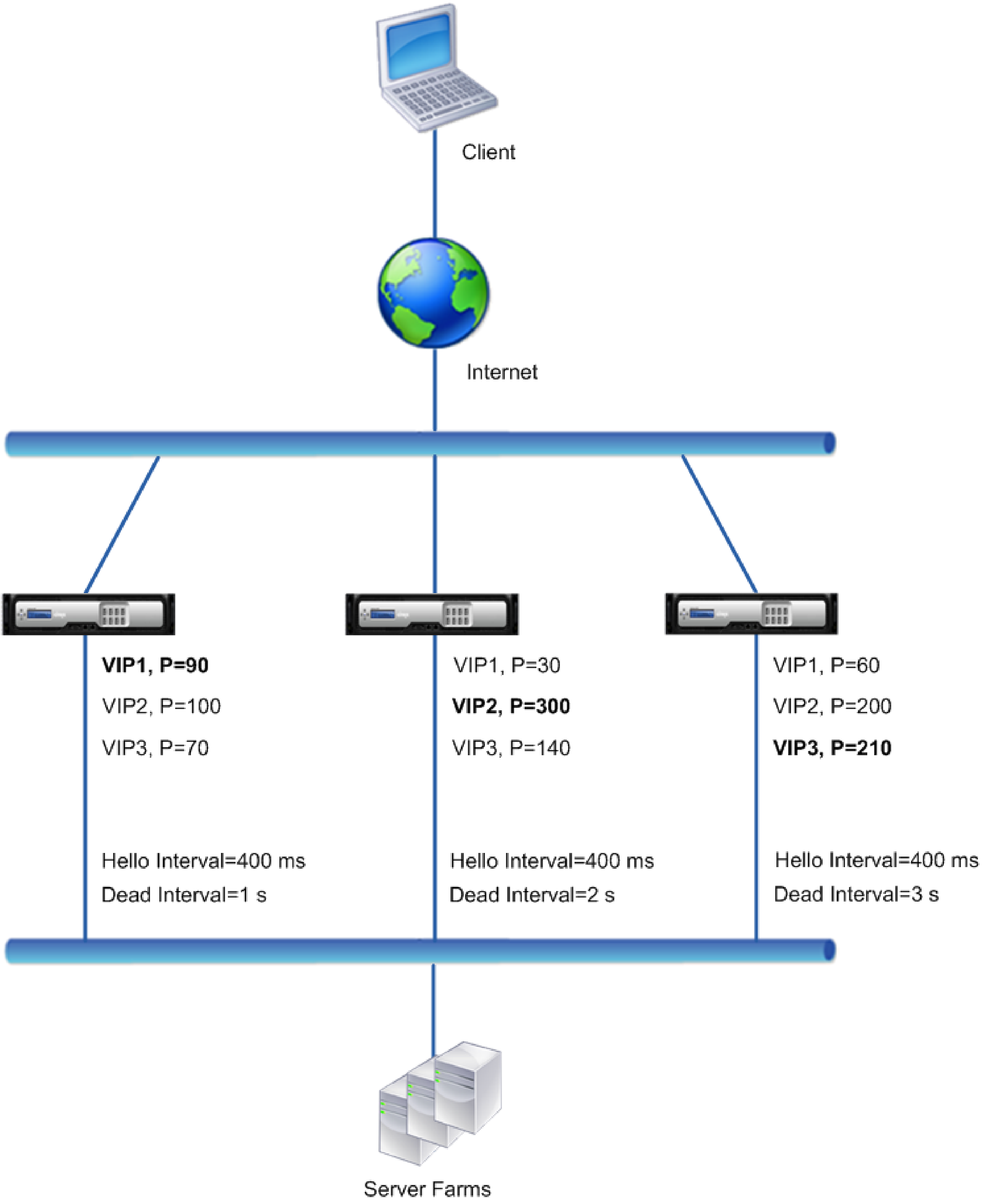
1. NS1 sends hello messages at a set hello interval of 400 ms to NS2 and NS3 for the VIP1 address, because VIP1 is active (the master) on NS1. Similarly, NS2 sends hello messages for VIP2, and NS3 sends hello messages for VIP3.

2. On NS1, the set dead interval applies to VIP2 and VIP3, because they are inactive (backups) on NS1. Similarly, on NS2, the set dead interval applies to VIP1 and VIP3, and on NS3, the set dead interval applies to VIP1 and VIP2.
3. If NS1 goes down, NS2 and NS3 consider NS1 to be down if they receive no hello messages from NS1 for 2 seconds (the dead interval). VIP1 on NS3 takes over and becomes active (master) because its VRID priority (60) is higher than that of VIP1 of NS2 (30).

Example 2: Nodes with Different VRRP Dead Intervals

Consider a VRRP deployment similar to the deployment described in Example1 but with a different dead interval on each node (NS1, NS2, and NS3). Preemption is disabled for the VIP addresses in all the three nodes.

The following table lists the settings used in this example: VRRP interval example 2 settings.


Client
Internet
VIP1, P=90
VIP2, P=100
VIP3, P=70
Hello Interval=400 ms
Dead Interval=1 s
VIP1, P=30
VIP2, P=300
VIP3, P=140
Hello Interval=400 ms
Dead Interval=2 s
VIP1, P=60
VIP2, P=200
VIP3, P=210
Hello Interval=400 ms
Dead Interval=3 s
Server Farms

The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to be down after not receiving any hello messages from NS1 for 2 seconds (NS2's dead interval).
2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages

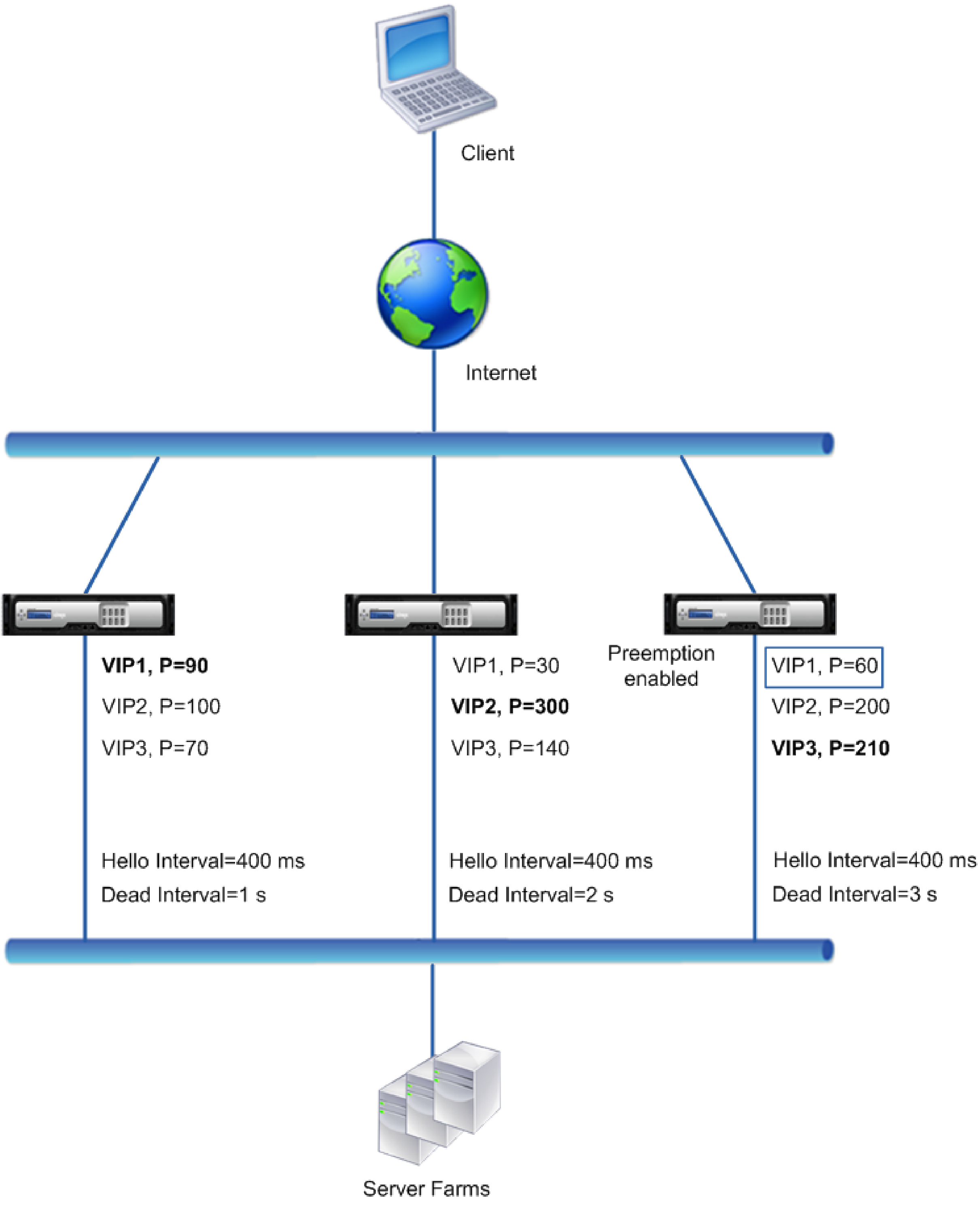
for VIP1.

Even though VIP1 on NS3 has a higher VRIP priority (60) than does VIP1 on NS2 (30), NS3's larger dead interval (3 seconds, vs. 2 seconds for NS2), prevents VIP1 on NS3 from taking over before VIP 1 on NS2 has already done so.

Example 3: Nodes with Different Dead Intervals and Preemption Enabled

Consider a VRRP deployment similar to the deployment described in Example1 but with different dead intervals on the three nodes, NS1, NS2, and NS3, and with preemption enabled for the VIP1 address on NS3.

The following table lists the settings used in this example: VRRP interval example 3 settings.


Client
Internet
VIP1, P=90
VIP2, P=100
VIP3, P=70
Hello Interval=400 ms
Dead Interval=1 s
VIP1, P=30
VIP2, P=300
VIP3, P=140
Hello Interval=400 ms
Dead Interval=2 s
Preemption enabled
VIP1, P=60
VIP2, P=200
VIP3, P=210
Hello Interval=400 ms
Dead Interval=3 s
Server Farms

The execution flow is as follows when NS1 goes down:

1. NS2 considers NS1 to down after not receiving any hello messages from NS1 for 2 seconds (NS2's set dead interval). At this time, NS3, with a dead interval of 3 seconds, does not consider NS1 to be down.

2. VIP1 on NS2 takes over and becomes active (master). NS2 now starts sending hello messages for VIP1.
3. Upon receiving hello messages from NS2 for VIP1, NS3 preempts NS2 for VIP1 because preemption is enabled for VIP1 of NS3 and the VRID priority (60) of VIP1 of NS3 is higher than that (30) of VIP1 of NS2.
4. VIP1 on NS3 takes over and becomes active (master). NS3 now starts sending hello messages for VIP1.

Configuring Health Tracking based on Interface State

September 21, 2020

To ensure that a backup VIP address takes over as the master VIP before the node of the current master VIP address goes down completely, you can configure a node to change the priority of a VIP address when the state of an interface on the node changes. For example, the node reduces the priority of a VIP address when the state of an interface changes to DOWN, and increases the priority when the state of the interface changes to UP. This feature is a per node configuration for each VIP address.

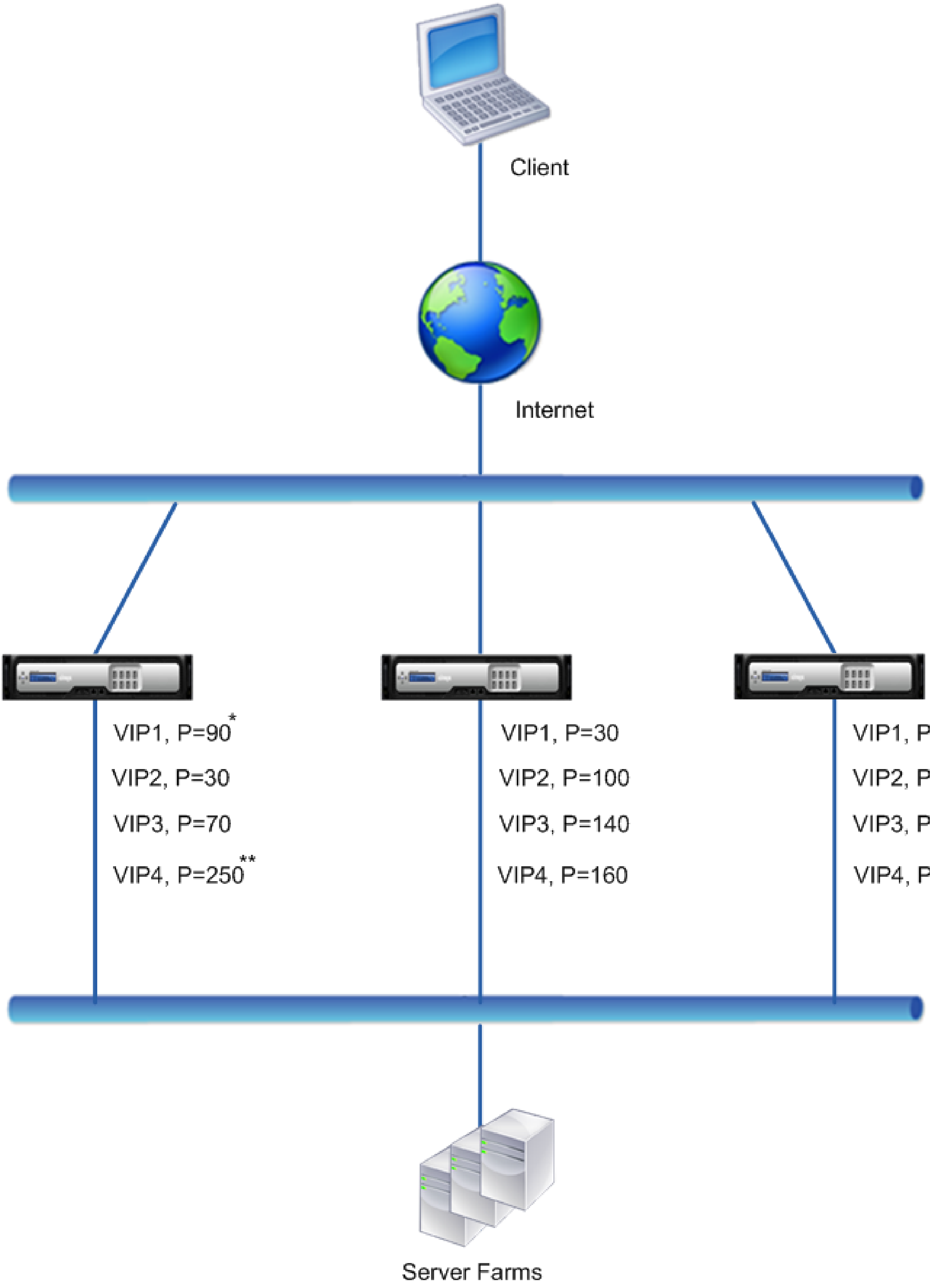
Example

Consider an active-active deployment consisting of NetScaler appliances NS1, NS2, and NS3. Virtual IP addresses VIP1, VIP2, VIP3, and VIP4 are configured on each of these appliances. Because of their priorities, VIP1 and VIP4 are active on NS1, VIP2 is active on NS2, and VIP3 is active on NS3.

To ensure that the active VIP addresses on NS1 are taken over by either NS2 or NS3 before NS1 goes down completely, interface-based health tracking is configured for the VIP1 and VIP4 addresses on NS1. Configuring interface-based health tracking for a VIP address includes associating the desired interfaces and setting the reduced priority (trackifNumPriority) parameter for the associated VRID of the VIP address. For example, on NS1, interfaces 1/2, 1/3, and 1/5 are associated to the VRID of VIP1, and reduced priority is set to 20.

Preemption is enabled for these VIP addresses in all three nodes.

The following table lists the settings used in this example: Health tracking example settings.


Client
Internet
VIP1, P=90*
VIP2, P=30
VIP3, P=70
VIP4, P=250**
VIP1, P=30
VIP2, P=100
VIP3, P=140
VIP4, P=160
VIP1, P=60
VIP2, P=200
VIP3, P=210
VIP4, P=150
Server Farms

* Packet Interfaces = 1/2, 1/3, 1/5
Reduced Priority = 20

** Packet Interfaces = 1/5, 1/7
Reduced Priority = 55

The execution flow is as follows on NS1 when multiple interface on NS1 goes down:

1. If interface 1/3 goes down, the priority of address VIP1 is reduced by 20 (VIP1's reduced priority value), because interface 1/3 is associated with VIP1:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (90-20) = 70
2. Similarly, if interface 1/5 goes down, the priority of address VIP1 is further reduced:
   - Effective priority of VIP1 = (Current priority - reduced priority) = (70-20) = 50
3. At this point, the effective priority of VIP1 on NS1 is less than the priority of VIP1 on NS3. NS3 preempts NS1 for VIP1. VIP1 on NS3 takes over and becomes active (master).
4. Also, because interface 1/5 is also associated with VIP4, the priority of VIP4 is reduced by the VIP4's reduced priority value (55).
   - Effective priority of VIP4 = (250 - 55) = 195
5. If interface 1/7 goes down, the priority of VIP4 is further reduced:
   - Effective priority of VIP4 = (Current priority - reduced priority) = (195-55) = 145
6. At this point, the effective priority of VIP4 on NS1 is less than the priority of VIP4 on NS2. NS2 preempts NS1 for VIP4. VIP4 on NS3 takes over and becomes active (master). This configuration ensures that none of the four VIP addresses are active on NS1 before it completely goes down.

Configuration Steps for IPv4 Active-Active Mode

To configure this feature on a node for a VIP address, you set the Reduced Priority (trackifNumPriority) parameter, and then associate the interfaces whose state is to be tracked for changing the priority of the VIP address. When any of the associated interface's state changes to DOWN or UP, the node reduces or increases the priority of the VIP address by the configured Reduced Priority (trackifNumPriority) value.

To set reduced priority and bind interfaces to the virtual router ID by using the NetScaler command line:

At the command prompt, type:

- **set vrID** <id> [-**trackifNumPriority** <positive_integer>]
- **bind vrID** <id> -**trackifNum** <interface_name>
- **show vrID** <id>

Example:

```
    > set vrID 125 -trackifNumPriority 10
    Done

    > bind vrID 125 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

To set reduced priority and bind interfaces to the virtual router ID by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC.**
2. On the **VMACs** tab, select a virtual router ID, and click **Edit**.
3. Under **Configure VMAC**, set the **Reduced Priority** parameter.
4. Select the **Interfaces tracked for the VRID** option and, under **Associate Interfaces**, add interfaces to the virtual router ID.

Configuration Steps for IPv6 Active-Active Mode

To configure this feature on a node for a VIP6 address, you set the Reduced Priority (trackifNumPriority) parameter, and then associate the interfaces whose state is to be tracked for changing the priority of the VIP6 address. When any of the associated interface's state changes to DOWN or UP, the node reduces or increases the priority of the VIP6 address by the configured Reduced Priority (trackifNumPriority) value.

To change the priority of a VIP address automatically by using the NetScaler command line:

At the command prompt, type one of the following sets of commands.

- If adding a new Virtual MAC6:
  - **add vrID6** <id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>
- If reconfiguring an existing Virtual MAC6:
  - **set vrID6** <id> [-**trackifNumPriority** <positive_integer>]
  - **bind vrID6** <id> -**trackifNum** <interface_name>
  - **show vrID6** <id>

Example:

```
    > set vrID6 130 -trackifNumPriority 10
    Done

    > bind vrID6 130 -trackifNum 1/4 1/5
    Done
<!--NeedCopy-->
```

Delaying Preemption

September 21, 2020

By default, a backup VIP address preempts the master VIP address immediately after its priority becomes higher than that of the master VIP. When configuring a backup VIP address, you can specify an amount of time by which to delay the preemption. Preemption delay time is a per-node setting for each backup VIP address.

The preemption delay setting for a backup VIP does not apply in the following conditions:

- The node of the master VIP goes down. In this case, the backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.
- The priority of the master VIP is set to zero. The backup VIP takes over as the master VIP after the dead interval set on the backup VIP's node.

Example: Delaying Preemption

Consider an active-active deployment consisting of NetScaler appliances NS1 and NS2. Virtual IP address VIP1 is configured on each of these appliances. Because of their priorities, VIP1 is master on NS2. Preemption is enabled and preemption delay time is set for VIP1 on these two nodes.

The following table lists the settings used in this example.

Entity and Parameters	Settings on NS1	Settings on NS2
VIP1 (for reference purposes only)	**IP address:** 192.0.1.10, **VRID:** 10, **Priority:** 100, **Preemption:** Enabled, **Preemption delay time:** 1000 seconds	**IP address:** 192.0.1.10, **VRID: 10**, **Priority:** 200, **Preemption:** Enabled, **Preemption delay time:** 2000 seconds
Dead Interval	1 Seconds	2 Seconds

Following are some examples of possible preemption behavior in this setup:

- If the priority of VIP1 on NS1 is set to a value (for example, 210) higher than that of VIP1 on NS2, VIP1 on NS1 takes over as master after its set preemption delay time (1000 secs).
- If a third node NS3 with the following VRRP settings is added to this deployment, VIP1 on NS3 becomes master after its set preemption delay time (3000 secs).
  - VIP1
    - VRID: 30
    - IP address:
    - Priority = 300
    - Preemption delay time = 3000 seconds
- If NS2 goes down, VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.

- If NS2 goes down and NS1 restarts, VIP1 on NS1 becomes master 1 second (set dead interval on NS1) after NS1 comes up. Preemption delay time for VIP1 on NS1 does not apply in this case.
- If the priority of VIP1 on NS2 is set to zero, VIP1 goes to standby mode. VIP1 on NS1 takes over as master after 1 second (set dead interval on NS1). Preemption delay time for VIP1 on NS1 does not apply in this case.

Configuring Delay Preemption for IPv4 Active-Active Mode

To configure preemption delay time for a VIP address, you set the preemption delay timer parameter of the associated virtual MAC address. You can the set this parameter when you add the address, or you can modify an existing virtual MAC address.

To configure preemption delay time by using the NetScaler command line:

- To set the preemption delay time while adding a Virtual MAC, at the command prompt, type:
  - **add vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**
- To set the preemption delay time while modifying a Virtual MAC, at the command prompt, type:
  - **set vrID** <id> -**preemptiondelaytimer** <secs>
  - **show vrID**

To configure preemption delay time by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab. While adding a new Virtual MAC, or editing an existing Virtual MAC, set the **Preemption Delay Timer** parameter.

Sample configuration:

The following configuration uses the settings listed in table in section Example: Delaying Preemption.

```
Settings on NS1

> set vrid param  - deadInterval 1

Done

> add ns ip 192.0.1.10 255.255.255.255  - type VIP

Done

> add vrid 10  - Priority 100  - Preemption Enable  -
    preemptiondelaytimer 1000

Done
```

```

    > bind ns ip 192.0.1.10 255.255.255.255 - vrid 10

    Done

    Settings on NS2

    > set vrid param - deadInterval 2

    Done

    > add ns ip 192.0.1.10 255.255.255.255 - type VIP

    Done

    > add vrid 20 - Priority 200 - Preemption Enable -
       preemptiondelaytimer 2000

    Done

    > set ns ip 192.0.1.10 255.255.255.255 - vrid 10

    Done
<!--NeedCopy-->
```

Configuring Delay Preemption for IPv6 Active-Active Mode

To configure preemption delay time for a VIP6 address, you set the preemption delay timer parameter of the associated Vitrtual MAC6 address. You can the set this parameter when you add the Vitrtual MAC6 address, or you can modify an existing Vitrtual MAC6 address.

To configure preemption delay time by using the NetScaler command line:

- To set the preemption delay time while adding a Virtual MAC6, at the command prompt, type:
  - **add vrID6 **<id> -**preemptiondelaytimer** <secs>
  - **show vrID6**
- To set the preemption delay time while modifying a Virtual MAC6, at the command prompt, type:
  - **set vrID6** <id> -**preemptiondelaytimer** <secs>
  - **show vrID6**

To configure preemption delay time by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC**.

2. On the **VMAC6** tab. While adding a Vitrtual MAC6 address, or editing an existing Vitrtual MAC6 address, set the **Preemption Delay Timer** parameter.

Keeping a VIP Address in the Backup State

September 21, 2020

You can force a VIP address to always stay in backup state. This operation is helpful in maintenance or testing of a VRRP deployment.

When a VIP address is forced to stay in backup state, it does not participate in VRRP state transitions. Also, it cannot become master even if all other nodes go down.

To force a VIP address to stay in backup state, you set the priority of the associated virtual MAC address to zero. To ensure that none of the VIP addresses of a node handle traffic during a maintenance process on the node, set all the priorities to zero.

You can the set the priority of a virtual MAC address while adding or modifying the address.

To force a VIP address to stay in the backup state by using the NetScaler command line:

- To set the priority while adding a Virtual MAC, at the command prompt, type:
  - **add vrID** <id> -**priority** 0
  - **show vrID**
- To set the priority while modifying a Virtual MAC, at the command prompt, type:
  - **set vrID** <id> -**priority** 0
  - **show vrID**

To force a VIP address to stay in backup state by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **VMAC**.
2. On the **VMAC** tab, while adding a new virtual MAC or editing an existing Virtual MAC, set the **Priority** parameter to zero.

Network Visualizer

September 21, 2020

The network visualizer shows a graphical view of all the interfaces, channels, VLANs, IP addresses, and bindings to VLANs on a NetScaler appliance. An enabled interface or channel has a black label. A disabled interface or channel has a red label.

This complete picture of the appliance's network connections can be useful for detecting flaws in the network design and for optimizing the network. It can also help a new administrator easily understand the appliance's network configuration.

To open the Network Visualizer:

Navigate to **System** > **Network**. In **Monitor Connections**, click **Network Visualizer**.

Configuring Link Layer Discovery Protocol

September 21, 2020

The NetScaler appliance supports the industry standard (IEEE 802.1AB) Link Layer Discovery Protocol (LLDP). LLDP is a layer 2 protocol that enables the NetScaler appliance to advertise its identity and capabilities to the directly connected devices, and also learn the identity and capabilities of these neighbour devices.

> **Note:**
> Link Layer Discovery Protocol (LLDP) is supported only in Citrix ADC MPX platforms.

Using LLDP, the NetScaler appliance transmits and receives information in the form of LLDP messages known as LLDP packet data units (LLDPUs). An LLDPU is a sequence of type, length, value (TLV) information elements. Each TLV holds a specific type of information about the device that transmits the LLDPDU. The NetScaler appliance sends the following TLVs in each LLDPU:

- Chassis ID
- Port ID
- Time-to-live value
- System name
- System description
- Port description
- System capabilities
- Management address
- Port VLAN ID
- Link aggregation

Note: You cannot specify the TLVs to be sent in LLDP messages.

NetScaler interfaces support the following LLDP modes:

- **NONE**. The interface neither receives from nor transmits LLDP messages to the directly connected device.
- **TRANSMITTER**. The interface transmits LLDP messages to the directly connected device but does not receive LLDP messages from the directly connected device.

- **RECEIVER**. The interface receives LLDP messages from the directly connected device but does not transmit LLDP messages to the directly connected device.
- **TRANSCEIVER**. The interface transmits LLDP messages to and receives LLDP messages from the directly connected device.

The LLDP mode of an interface depends on the LLDP mode configured at the global and the interface levels. The following table shows the modes resulting from the available combinations of global- and interface-level settings: Interface and global level LLDP modes.

Note the following points related to LLDP messages transmitted or received by the NetScaler:

- **Transmitting LLDP messages.** The NetScaler appliance transmits LLDPUs from interfaces that are operating in either TRANSMITTER or TRANSCEIVER LLDP mode.

  Following are the global LLDP transmitting parameters on the NetScaler appliance:

  - **Timer**. Interval, in seconds, between LLDPUs that the NetScaler appliance sends to a directly connected device.
  - **Holdtime Multiplier**. A multiplier for calculating the duration for which the receiving device stores the LLDP information in its database before discarding or removing it. The duration is calculated as the **Holdtime Multiplier** parameter value multiplied by the Timer parameter value.

- **Receiving LLDP Messages.** The NetScaler appliance stores the LLDPDU information in its Management Information base (MIB). The stored LLDP information is classified or grouped under the ID of the interface that received the LLDPDU. The NetScaler appliance retains this LLDP information for the duration specified in the received LLDPU.

  If the appliance receives another LLDPDU on an interface before the stored LLDP information for that interface is discarded, the appliance replaces the stored LLDP information for that interface with information in the new LLDPDU.

Configuration Steps

Configuring LLDP on a NetScaler appliance consists of the following tasks:

1. **Set global level LLDP parameters.** In this task, you set the global LLDP parameters such as LLDP Timer, Hold Time Multiplier, and LLDP mode.
2. **Set the interface level LLDP parameters.** In this task, you set the LLDP mode for an interface.
3. **(Optional) Display neighbor-device information.** You can display the neighbor-device LLDP information collected on all of the NetScaler appliance's interfaces, or just the LLDP information collected on specified interfaces. If you do not specify an interface, the information is shown for all interfaces.

Following are the prerequisites for configuring LLDP on a NetScaler appliance:

1. Make sure that you understand the standard LLDP protocol (IEEE 802.1AB).
2. Verify that you have configured LLDP on the desired directly connected devices.

NetScaler command line procedures

To set global level LLDP parameters by using the NetScaler command line:

At the command prompt, type:

- set lldp param [-[-holdtimeTxMult <positive_integer>][-timer <positive_integer>] [-Mode <Mode>]
- show lldp param

To configure an interface for LLDP by using the NetScaler command line:

At the command prompt, type:

- set interface <id> -lldpmode <lldpmode>
- show interface <id>

To display neighbor device information by using the NetScaler command line:

At the command prompt, type one of the following commands:

- show lldp neighbors
- show lldp neighbors <ifnum>

NetScaler GUI procedures

To set the global level LLDP parameters by using the NetScaler GUI:

1. Navigate to System > Network, and click Configure LLDP Parameters.
2. Set the following parameters:
    - Hold Timer Multiplier
    - Timer
    - Mode

To configure an interface for LLDP by using the NetScaler GUI:

Navigate to System > Network > Interfaces, open the interface, and set the LLDP mode parameter.

To display neighbor device information by using the NetScaler GUI:

Navigate to System > Network > Interfaces and, in the Action list, select View LLDP Neighbors.

LLDP Support in a Cluster Setup

In a cluster setup, the NetScaler GUI and NetScaler CLI display the LLDP neighbour configuration of all or specific cluster nodes when the GUI or CLI is accessed through the Cluster IP address (CLIP). Any change made to the global level LLDP mode is applied to the global level LLDP mode on each of the cluster nodes.

Consider an example of a cluster setup of three nodes, NS1, NS2, and NS3. Each of these nodes are connected to both routers Router-1 and Router-2. The following output is displayed when the **show lldp neighbor -summary** operation is performed on the Cluster CLI that is accessed through the Cluster IP address (CLIP) of the cluster setup. The output shows the LLDP neighbour information of all these nodes.

```
> show lldp neighbor -summary

Node Id: 1
------------------------------------------------------------
    Interface    ChassisId           PortId     System name
------------------------------------------------------------
1     1/1/1      fe:c7:3b:13:bd:11    1/1         Router-1

2     1/1/2      12:68:7b:9e:4c:11    1/1         Router-2

Node Id: 2
-------------------------------------------------------------
    Interface    ChassisId           PortId     System name
-------------------------------------------------------------
1     2/1/1      fe:c7:3b:13:bd:12    1/2        Router-1

2     2/1/2      12:68:7b:9e:4c:12    1/2        Router-2

Node Id: 3
-------------------------------------------------------------
    Interface    ChassisId            PortId     System name
-------------------------------------------------------------

1     3/1/1       fe:c7:3b:13:bd:13    1/3        Router-1

2     3/1/2       12:68:7b:9e:4c:13    1/3        Router-2

Done
<!--NeedCopy-->
```

Jumbo Frames

September 21, 2020

NetScaler appliances support receiving and transmitting jumbo frames containing up to 9216 bytes of IP data. Jumbo frames can transfer large files more efficiently than it is possible with the standard IP MTU size of 1500 bytes.

A NetScaler appliance can use jumbo frames in the following deployment scenarios:

- Jumbo to Jumbo. The appliance receives data as jumbo frames and sends it as jumbo frames.
- Non-Jumbo to Jumbo. The appliance receives data as regular frames and sends it as jumbo frames.
- Jumbo to Non-Jumbo. The appliance receives data as jumbo frames and sends it as regular frames.

The NetScaler appliance supports jumbo frames in a load balancing configuration for the following protocols:

- TCP
- Any protocol over TCP (for example, HTTP)
- SIP
- RADIUS

Configuring Jumbo Frames Support on a NetScaler Appliance

September 21, 2020

To enable the NetScaler appliance to support jumbo frames, you set the MTU to more than 1500 on interfaces or LA channels, and on VLANs on which you want the NetScaler appliance to support jumbo frames.

Points to consider before setting the MTU of interfaces, LA channels, or VLANs on a NetScaler appliance

1. When you create an LA channel, the channel takes the MTU of the first bound interface if no MTU is specified for the channel.
2. The MTU for a channel is propagated to all the bound interfaces.
3. When an interface is bound to the channel whose MTU is different from the interface's MTU, the interface goes onto the inactive list.
4. When you change the MTU of a member interface, the interface goes onto the inactive list.

5. When an interface is unbound from the channel, the interface retains the MTU value of the channel.
6. You can set the MTU for an interface, channel, or VLAN to a value in the range of 1500-9216.
7. You cannot set the MTU on the default VLAN. The NetScaler appliance uses the MTU of the interface through which it receives or sends data from or to the default VLAN.
8. For TCP based traffic on a load balancing configuration on a NetScaler appliance, MSSs are set accordingly at each end point for supporting jumbo frames:
    - For a connection between a client and a load balancing virtual server on the NetScaler appliance, the MSS on the NetScaler appliance is set in a TCP profile, which is then bound to the load balancing virtual server.
    - For a connection between the NetScaler appliance and a server, the MSS on NS1 is set in a TCP profile, which is then bound to the service representing the server on the NetScaler appliance.
    - By default, a TCP profile nstcp_default_profile is bound to all TCP based load balancing servers and services on the NetScaler appliance.
    - For supporting jumbo frames, you can either change the MSS value of the TCP profile nstcp_default_profile, or create a custom TCP profile and set its MSS accordingly, and then bind the custom TCP profile to the desired load balancing virtual servers and services.
    - The default MSS value of any TCP profile is 1460.

NetScaler command line procedures

To set the MTU of an interface by using the NetScaler command line:

At the command prompt, type:

- set interface <id> -mtu <positive_integer>
- show interface <id>

Example:

```
> set interface 10/1  - mtu 9000
Done
<!--NeedCopy-->
```

To set the MTU of a channel by using the NetScaler command line:

At the command prompt, type:

- `set channel <id> -mtu <positive_integer>`
- `show channel <id>`

Example:

```
> set channel LA/1 -mtu 9000
Done
<!--NeedCopy-->
```

To set the MTU of a VLAN by using the NetScaler command line:

At the command prompt, type:

- add vlan <id> -mtu <positive_integer>
- show vlan <id>

Example:

```
> set vlan 20 -mtu 9000
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To set the MTU of an interface by using the NetScaler GUI:

Navigate to System > Network > Interfaces, open the interface, and set the Maximum Transmission Unit parameter.

To set the MTU of a channel by using the NetScaler GUI:

Navigate to System > Network > Channels, open the channel, and set the Maximum Transmission Unit parameter.

To set the MTU of a VLAN by using the NetScaler GUI:

Navigate to System > Network > VLANs, open the VLAN, and set the Maximum Transmission Unit parameter.

Use Case 1 – Jumbo to Jumbo Setup

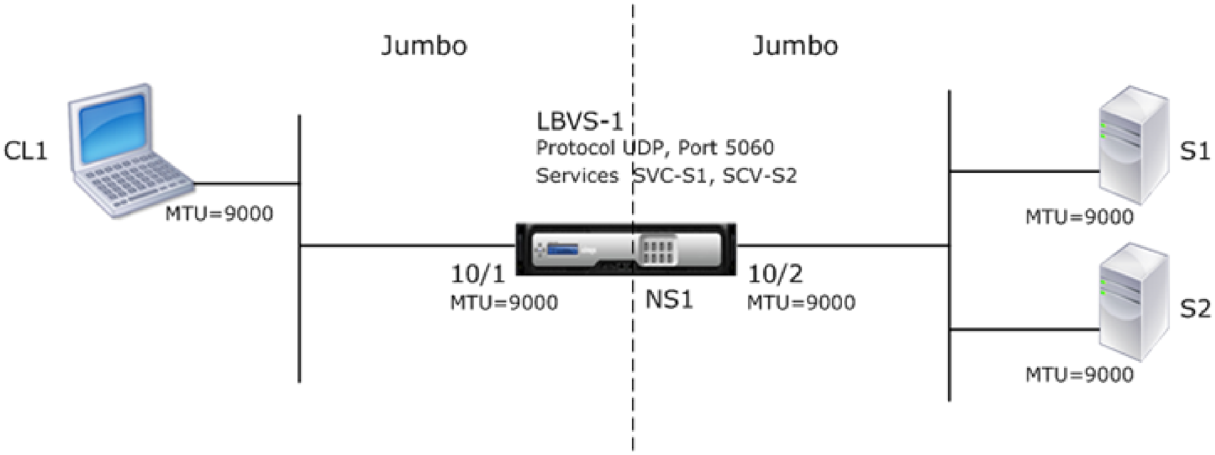
September 21, 2020

Consider an example of a jumbo to jumbo setup in which SIP load balancing virtual server LBVS-1, configured on NetScaler appliance NS1, is used to load balance SIP traffic across servers S1 and S2. The connection between client CL1 and NS1, and the connection between NS1 and the servers support jumbo frames.

Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2. Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively.

For supporting jumbo frames, the MTU is set to 9216, on NS1, for interfaces 10/1, 10/2, and VLANs VLAN 10, VLAN 20.

All other network devices, including CL1, S1, S2, in this setup example are also configured for supporting jumbo frames.


Jumbo
Jumbo
CL1
MTU=9000
LBVS-1
Protocol UDP, Port 5060
Services SVC-S1, SCV-S2
10/1
MTU=9000
NS1
10/2
MTU=9000
S1
MTU=9000
S2
MTU=9000

The following table lists the settings used in the example.

Entity	Name	Details
IP address of client CL1	-	192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	9000
	10/2	9000
	VLAN 10	9000
	VLAN 20	9000
Services on NS1 representing servers	SVC-S1	**IP address:** 198.51.100.19, **Protocol:** SIP, **Port:** 5060
	SVC-S2	**IP address:** 198.51.100.20, **Protocol:** SIP, **Port:** 5060

Entity	Name	Details
Load balancing virtual server on VLAN 10	LBVS-1	**IP address:** 203.0.113.15, **Protocol:** SIP, **Port:** 5060, **Bound services:** SVC-S1, SVC-S2

Following is the traffic flow of CL1's request to NS1:

1. CL1 creates a 20000-byte SIP request to send to LBVS-1 of NS1.
2. CL1 sends the request data in IP fragments to LBVS-1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which CL1 sends these fragments to NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
    - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
    - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068
3. NS1 receives the request IP fragments at interface 10/1. NS1 accepts these fragments, because the size of each of these fragments is equal to or less than the MTU (9000) of interface 10/1.
4. NS1 reassembles these IP fragments to form the 20000-byte SIP request. NS1 processes this request.
5. LBVS-1's load balancing algorithm selects server S1.
6. NS1 sends the request data in IP fragments to S1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/2, from which NS1 sends these fragments to S1. The IP packets are sourced with a SNIP address of NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
    - Size of the second IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
    - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 2048] = 2068

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates a 30000-byte SIP response to send to the SNIP address of NS1.
2. S1 sends the response data in IP fragments to the SNIP address of NS1. The size of each IP fragment is either equal to or less than the MTU (9000) set on the interface from which S1 sends these fragments to NS1.
    - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000

   - Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088

3. NS1 receives the response IP fragments at interface 10/2. NS1 accepts these fragments, because the size of each fragment is equal to or less than the MTU (9000) of interface 10/2.

4. NS1 reassembles these IP fragments to form the 30000-byte SIP response. NS1 processes this response.

5. NS1 sends the response data in IP fragments to CL1. The size of each IP fragment is either equal or less than the MTU (9000) of the interface 10/1, from which NS1 sends these fragments to CL1. The IP fragments are sourced with LBVS-1's IP address.

   - Size of the first IP fragment = [IP header + UDP header + SIP data segment] = [20 + 8 + 8972] = 9000
   - Size of the second and third IP fragment = [IP header + SIP data segment] = [20 + 8980] = 9000
   - Size of the last IP fragment=[IP header + SIP data segment] = [20 + 3068] = 3088

Configuration Tasks

The following table list the tasks, NetScaler commands, and examples for creating the required configuration on the NetScaler appliance.

Task	NetScaler Command Syntax	Example
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 9000 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 9000 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP

Task	NetScaler Command Syntax	Example
Create services representing SIP servers	add service <serviceName> <ip> SIP_UDP <port>, show service <name>	add service SVC-S1 198.51.100.19 SIP_UDP 5060 add service SVC-S2 198.51.100.20 SIP_UDP 5060
Create SIP load balancing virtual servers and bind the services to it	add lb vserver <name> SIP_UDP <ip> <port> bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 SIP_UDP 203.0.113.15 5060 bind lb vserver LBVS-1 SVC-S1 bind lb vserver LBVS-1 SVC-S2
Save the configuration	save ns config, show ns config	

Use Case 2 – Non-Jumbo to Jumbo Setup

September 21, 2020

Consider an example of a regular to jumbo setup in which load balancing virtual server LBVS-1, configured on a NetScaler appliance NS1, is used to load balance traffic across servers S1 and S2. The connection between client CL1 and NS1 supports regular frames, and the connection between NS1 and the servers supports jumbo frames.

Interface 10/1 of NS1 receives or sends traffic from or to client CL1. Interface 10/2 of NS1 receives or sends traffic from or to server S1 or S2.

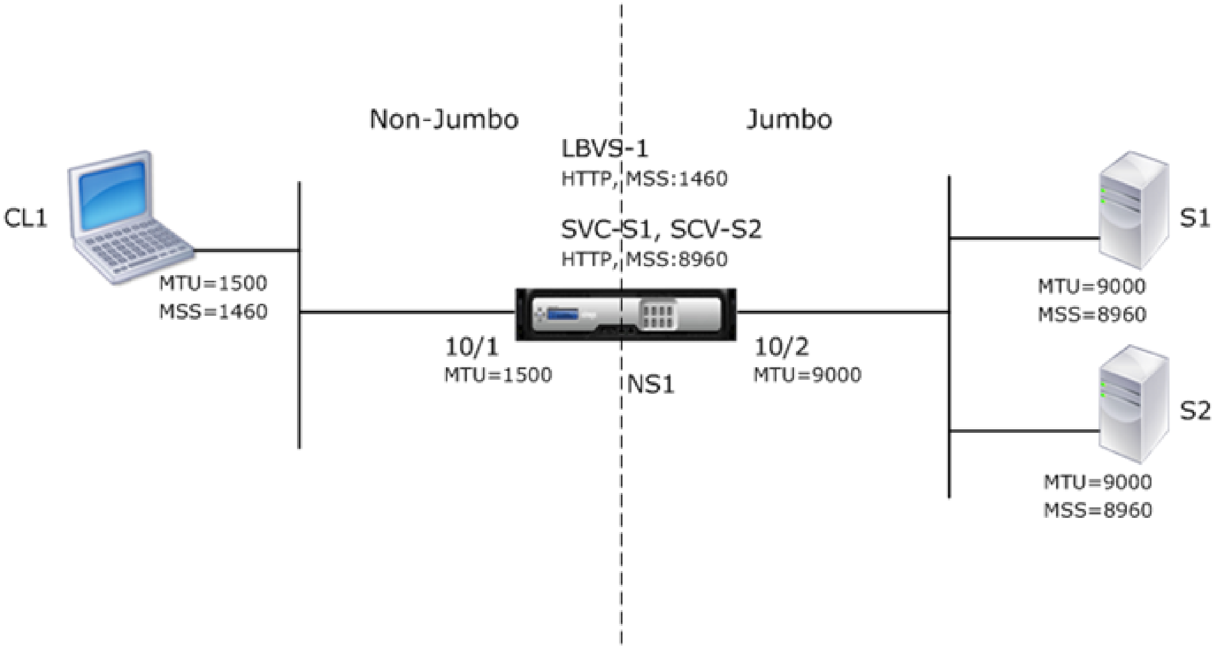
Interfaces 10/1 and 10/2 of NS1 are part of VLAN 10 and VLAN 20, respectively. For supporting only regular frames between CL1 and NS1, the MTU is set to the default value of 1500 for both interface 10/1 and VLAN 10

For supporting jumbo frames between NS1 and the servers, the MTU is set to 9000 for interface 10/2 and VLAN 20. Servers and all other network devices between NS1 and the servers are also configured for supporting jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For supporting jumbo frames for the connection between a SNIP address of NS1 and S1 or S2, the MSS on NS1 is set accordingly in a custom TCP profile, which is bound to the services (SVC-S1 and SVC-S1) representing S1 and S2 on NS1.
- For supporting only regular frames for the connection between CL1 and virtual server LBVS-1 of NS1, default TCP profile nstcp_default_profile is used that is by default bound to LBVS-1 and

has the MSS set to the default value of 1460.


Non-Jumbo
Jumbo
LBVS-1
HTTP, MSS:1460
SVC-S1, SCV-S2
HTTP, MSS:8960
CL1
MTU=1500
MSS=1460
10/1
MTU=1500
NS1
10/2
MTU=9000
S1
MTU=9000
MSS=8960
S2
MTU=9000
MSS=8960

The following table lists the settings used in this example.

Entity	Name	Details
IP address of client CL1		192.0.2.10
IP address of servers	S1	198.51.100.19
	S2	198.51.100.20
SNIP address on NS1		198.51.100.18
MTU specified for interfaces and VLANs on NS1	10/1	1500
	10/2	9000
	VLAN 10	1500
	VLAN 20	9000
Default TCP profile	nstcp_default_profile	MSS:1460
Custom TCP profile	NS1-SERVERS-JUMBO	MSS: 8960
Services on NS1 representing servers	SVC-S1	IP address: 198.51.100.19, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)

Entity	Name	Details
	SVC-S2	IP address: 198.51.100.20, Protocol: HTTP, Port: 80, TCP profile: NS1-SERVERS-JUMBO (MSS: 8960)
Load balancing virtual server on VLAN 10	LBVS-1	IP address = 203.0.113.15, Protocol: HTTP, Port:80, Bound services: SVC-S1, SVC-S2, TCP Profile: nstcp_default_profile (MSS:1460)

Following is the traffic flow of CL1's request to S1 in this example:

1. Client CL1 creates a 200-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their respective TCP MSS values while establishing the connection.
3. Because NS1's MSS is larger than the HTTP request, CL1 sends the request data in a single IP packet to NS1.

   Size of the request packet = [IP Header + TCP Header + TCP Request] = [20 + 20 + 200] = 240
4. NS1 receives the request packet at interface 10/1 and then processes the HTTP request data in the packet.
5. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.
6. Because S1's MSS is larger than the HTTP request, NS1 sends the request data in a single IP packet to S1.

   Size of the request packet = [IP Header + TCP Header + [TCP Request] = [20 + 20 + 200] = 240

Following is the traffic flow of S1's response to CL1 in this example:

1. Server S1 creates an 18000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
   - Size of the first two packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000

    - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120

3. NS1 receives the response packets at interface 10/2.
4. From these IP packets, NS1 assembles all the TCP segments to form the HTTP response data of 18000 bytes. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets, from interface 10/1, to CL1. These IP packets are sourced from LBVS-1's IP address and destined to CL1's IP address.
    - Size of all packets except the last one = [IP Header + TCP Header + (TCP payload=CL1's MSS size)] = [20 + 20 + 1460] = 1500
    - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 480] = 520

Configuration Tasks

The following table list the tasks, NetScaler commands, and examples for creating the required configuration on the NetScaler appliance.

Tasks	NetScaler Command line Syntax	Examples
Set the MTU of the desired interfaces for supporting jumbo frames	set interface <id> -mtu <positive_integer>, show interface <id>	set int 10/1 -mtu 1500 set int 10/2 -mtu 9000
Create VLANs and set the MTU of the desired VLANs for supporting jumbo frames	add vlan <id> -mtu <positive_integer>, show vlan <id>	add vlan 10 -mtu 1500 add vlan 20 -mtu 9000
Bind interfaces to VLANs	bind vlan <id> -ifnum <interface_name>, show vlan <id>	bind vlan 10 -ifnum 10/1 bind vlan 20 -ifnum 10/2
Add a SNIP address	add ns ip <IPAddress> <netmask> -type SNIP, show ns ip	add ns ip 198.51.100.18 255.255.255.0 -type SNIP
Create services representing HTTP servers	add service <serviceName> <ip> HTTP <port>, show service <name>	add service SVC-S1 198.51.100.19 http 80, add service SVC-S2 198.51.100.20 http 80

Tasks	NetScaler Command line Syntax	Examples
Create HTTP load balancing virtual servers and bind the services to it	add lb vserver <name> HTTP <ip> <port>, bind lb vserver <vserverName> <serviceName>, show lb vserver <name>	add lb vserver LBVS-1 http 203.0.113.15 80, bind lb vserver LBVS-1 SVC-S1, bind lb vserver LBVS-1 SVC-S2
Create a custom TCP profile and set its MSS for supporting jumbo frames	add tcpProfile <name> -mss <positive_integer>, show tcpProfile <name>	add tcpprofile NS1-SERVERS-JUMBO -mss 8960
Bind the custom TCP profile to the desired services	set service <Name> -tcpProfileName <string>, show service <name>	set service SVC-S1 -tcpProfileName NS1-SERVERS-JUMBO, set service SVC-S2 -tcpProfileName NS1-SERVERS-JUMBO
Save the configuration	save ns config, show ns config	

Use Case 3 – Coexistence of Jumbo and Non-Jumbo flows on the Same Set of Interfaces

September 21, 2020

Consider an example in which load balancing virtual servers LBVS-1 and LBVS-2 are configured on NetScaler appliance NS1. LBVS-1 is used to load balance HTTP traffic across servers S1 and S2, and LBVS-2 is used to load balance traffic across servers S3 and S4.

CL1 is on VLAN 10, S1 and S2 are on VLAN20, CL2 is on VLAN 30, and S3 and S4 are on VLAN 40. VLAN 10 and VLAN 20 support jumbo frames, and VLAN 30 and VLAN 40 support only regular frames.

In other words, the connection between CL1 and NS1, and the connection between NS1 and server S1 or S2 support jumbo frames. The connection between CL2 and NS1, and the connection between NS1 and server S3 or S4 support only regular frames.

Interface 10/1 of NS1 receives or sends traffic from or to clients. Interface 10/2 of NS1 receives or sends traffic from or to the servers.

Interface 10/1 is bound to both VLAN 10 and VLAN 30 as a tagged interface, and interface 10/2 is bound to both VLAN 20 and VLAN 40 as a tagged interface.

For supporting jumbo frames, the MTU is set to 9216 for interfaces 10/1 and 10/2.

On NS1, the MTU is set to 9000 for VLAN 10 and VLAN 20 for supporting jumbo frames, and the MTU is set to the default value of 1500 for VLAN 30 and VLAN 40 for supporting only regular frames.

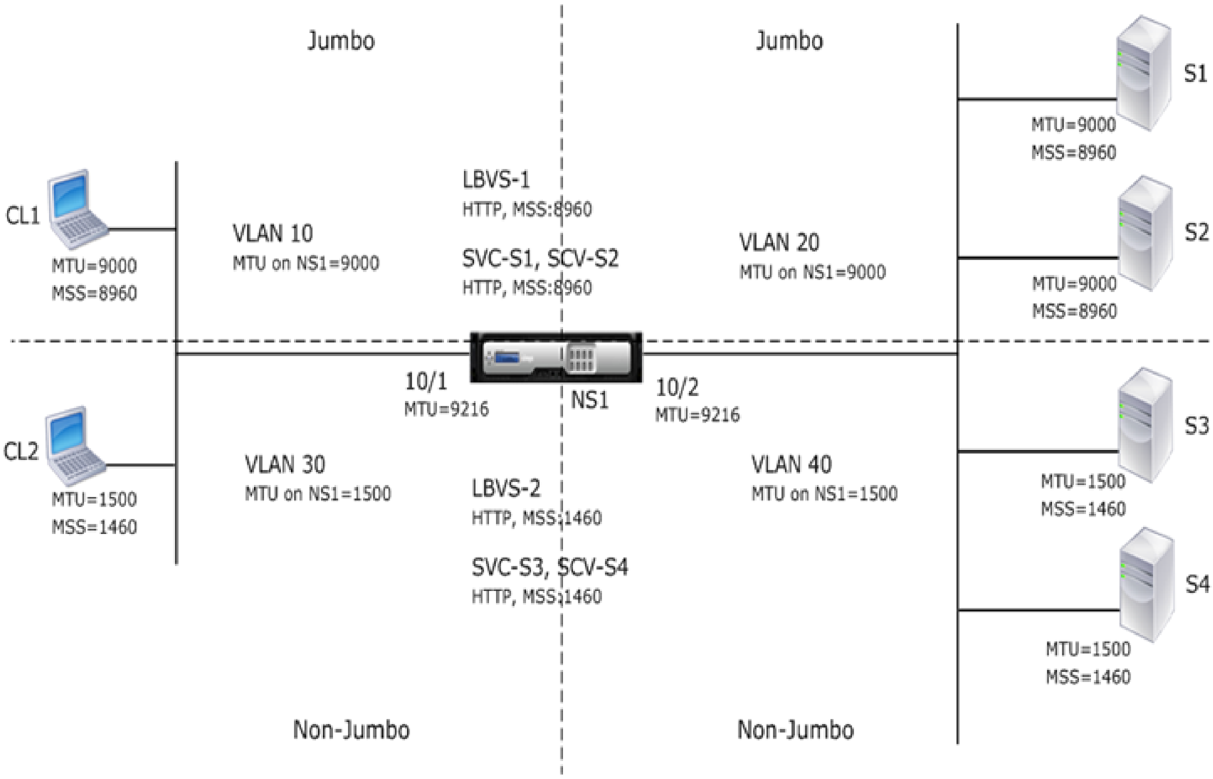
The effective MTU on a NetScaler interface for VLAN tagged packets is of the MTU of the interface or the MTU of the VLAN, whichever is lower. For example:

- The MTU of interface 10/1 is 9216. The MTU of VLAN 10 is 9000. On interface 10/1, the MTU of VLAN 10 tagged packets is 9000.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 20 is 9000. On interface 10/2, the MTU of VLAN 20 tagged packets is 9000.
- The MTU of interface 10/1 is 9216. The MTU of VLAN 30 is 1500. On interface 10/1, the MTU of VLAN 30 tagged packets is 1500.
- The MTU of interface 10/2 is 9216. The MTU of VLAN 40 is 1500. On interface 10/2, the MTU of VLAN 40 tagged packets is 9000.

CL1, S1, S2, and all network devices between CL1 and S1 or S2 are configured for jumbo frames.

Since HTTP traffic is based on TCP, MSSs are set accordingly at each end point for supporting jumbo frames.

- For the connection between CL1 and virtual server LBVS-1 of NS1, the MSS on NS1 is set in a TCP profile, which is then bound to LBVS-1.
- For the connection between a SNIP address of NS1 and S1, the MSS on NS1 is set in a TCP profile, which is then bound to the service (SVC-S1) representing S1 on NS1.


Jumbo
Jumbo
S1
MTU=9000
MSS=8960
LBVS-1
HTTP, MSS:8960
CL1
VLAN 10
MTU on NS1=9000
SVC-S1, SCV-S2
HTTP, MSS:8960
VLAN 20
MTU on NS1=9000
S2
MTU=9000
MSS=8960
MTU=9000
MSS=8960
MTU=9000
MSS=8960
10/1
MTU=9216
NS1
10/2
MTU=9216
S3
CL2
VLAN 30
MTU on NS1=1500
LBVS-2
HTTP, MSS:1460
VLAN 40
MTU on NS1=1500
MTU=1500
MSS=1460
MTU=1500
MSS=1460
SVC-S3, SCV-S4
HTTP, MSS:1460
S4
MTU=1500
MSS=1460
Non-Jumbo
Non-Jumbo

The following table lists the settings used in this example: Jumbo frames use case 3 example settings.

Following is the traffic flow of CL1's request to S1:

1. Client CL1 creates a 20000-byte HTTP request to send to virtual server LBVS-1 of NS1.
2. CL1 opens a connection to LBVS-1 of NS1. CL1 and NS1 exchange their TCP MSS values while establishing the connection.
3. Because NS1's MSS value is smaller than the HTTP request, CL1 segments the request data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 10 to NS1.
   - Size of the first two packets = [IP Header + TCP Header + (TCP segment=NS1 MSS)] = [20 + 20 + 8960] = 9000
   - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120
4. NS1 receives these packets at interface 10/1. NS1 accepts these packets because the size of these packets is equal to or less than the effective MTU (9000) of interface 10/1 for VLAN 10 tagged packets.
5. From the IP packets, NS1 assembles all the TCP segments to form the 20000-byte HTTP request. NS1 processes this request.
6. LBVS-1's load balancing algorithm selects server S1, and NS1 opens a connection between one of its SNIP addresses and S1. NS1 and CL1 exchange their respective TCP MSS values while establishing the connection.
7. NS1 segments the request data into multiples of S1's MSS and sends these segments in IP pack-

ets tagged as VLAN 20 to S1.

- Size of the first two packets = [IP Header + TCP Header + (TCP payload=S1 MSS)] = [20 + 20 + 8960] = 9000
- Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 2080] = 2120

Following is the traffic flow of S1's response to CL1:

1. Server S1 creates a 30000-byte HTTP response to send to the SNIP address of NS1.
2. S1 segments the response data into multiples of NS1's MSS and sends these segments in IP packets tagged as VLAN 20 to NS1. These IP packets are sourced from S1's IP address and destined to the SNIP address of NS1.
    - Size of first three packet = [IP Header + TCP Header + (TCP segment=NS1's MSS size)] = [20 + 20 + 8960] = 9000
    - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160
3. NS1 receives the response packets at interface 10/2. NS1 accepts these packets, because their size is equal to or less than the effective MTU value (9000) of interface 10/2 for VLAN 20 tagged packets.
4. From these IP packets, NS1 assembles all the TCP segments to form the 30000-byte HTTP response. NS1 processes this response.
5. NS1 segments the response data into multiples of CL1's MSS and sends these segments in IP packets tagged as VLAN 10, from interface 10/1, to CL1. These IP packets are sourced from LBVS's IP address and destined to CL1's IP address.
    - Size of first three packet = [IP Header + TCP Header + [(TCP payload=CL1's MSS size)] = [20 + 20 + 8960] = 9000
    - Size of the last packet = [IP Header + TCP Header + (remaining TCP segment)] = [20 + 20 + 3120] = 3160

Configuration Tasks

Following table lists tasks, commands, and examples for creating the required configuration on the NetScaler appliance: Jumbo frames use case 3 configuration tasks.

NetScaler Support for Microsoft Direct Access Deployment

September 21, 2020

Microsoft Direct Access is a technology that enables remote users to seamlessly and securely connect to the enterprise's internal networks, without the need to establish a separate VPN connection. Unlike VPN connections, which require user intervention to open and close connections, a Direct Access-enabled client connects automatically to the enterprise's internal networks whenever the client connects to the Internet.

Manage-Out is a Microsoft Direct Access feature that allows administrators inside the enterprise network to connect to Direct Access clients outside the network and manage them (for example, performing administration tasks, such as scheduling service updates, and providing remote support.

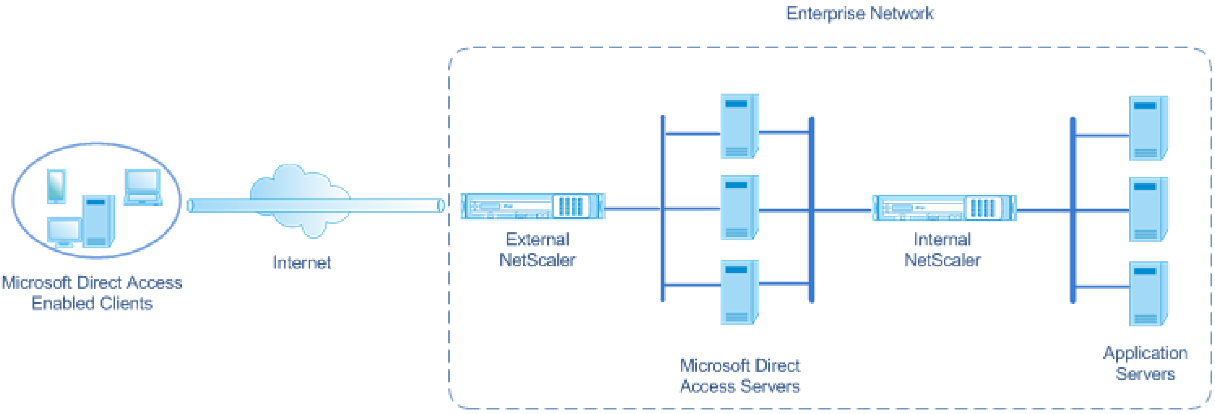
In a Direct Access deployment, NetScaler appliances provide high availability, scalability, high performance, and security. NetScaler load balancing functionality sends client traffic through the most appropriate server. The appliances can also forward Manage-Out traffic through the right path to reach the client.

Architecture

The architecture of a Microsoft Direct Access deployment consists of Direct Access enabled clients, Direct Access servers, application servers, and internal and external NetScaler appliances. Clients connect to an application server through a Direct Access server. An external NetScaler appliance load balance the client traffic to a Direct Access server, and an Internal NetScaler appliance forwards the client traffic from the Direct Access server to the destination application server. Direct Access is used for tunneling the client's IPv6 traffic over the IPv4 network. An IPv4 load balancing virtual server on the external NetScaler appliance load balances the client's tunneled traffic to one of the Direct Access servers. The Direct Access server extracts the IPv6 packets from the received client's IPv4 packets and sends them to the destination application server through the internal NetScaler appliance. The Internal NetScaler appliance has forwarding session rules with the source route cache option enabled for storing Layer 2 and Layer 3 connection information about the client's traffic from the Direct Access Server. The NetScaler appliance stores the following Layer 2 and Layer 3 information in a table called the source route cache table:

- Source IP address of the received packet
- MAC address of the Direct Access server that sent the packet
- VLAN ID of the NetScaler appliance that received the packet
- Interface ID of the NetScaler appliance that received the packet

The NetScaler appliance uses the information in the source route cache table for forwarding a response to the same Direct Access server because it has the tunneling information to reach the client. Also, the Internal NetScaler uses the source route cache table to forward application server's Manage-out traffic to the appropriate Direct Access server to reach a particular client.


Enterprise Network
Microsoft Direct Access Enabled Clients
Internet
External NetScaler
Microsoft Direct Access Servers
Internal NetScaler
Application Servers

Configuring the Internal NetScaler Appliance in a Microsoft Direct Access Deployment

To configure the Internal NetScaler appliance for forwarding an application server's response and manage-out traffic to the appropriate Direct Access Gateway, configure forwarding session rules. In each rule, set the sourceroutecache parameter to ENABLED.

To create a forwarding session rule by using the NetScaler command line:

At the command prompt, type:

- **add forwardingSession** <name> ((<network> [<netmask>]) | **-acl6name** <string> | **-aclname** <string>) **-sourceroutecache** (**ENABLED** | **DISABLED**]
- **show forwardingsession** <name>

Sample configuration:

In the following example, forwarding-session rule MS-DA-FW-1 is created on the internal NetScaler appliance. The forwarding session stores Layer 2 and Layer 3 information for any incoming IPv6 packets from a Direct Access server that matches source IPv6 prefix 2001:DB8::/96.

```
> add forwardingSession MS-DA-FW-1 2001:DB8::/96 -sourceroutecache -
  ENABLED
Done
```

Displaying the Source Route Cache Table

You can display the source route cache table for monitoring or detecting any unwanted connections between direct access servers and application servers.

To display the source route cache table by using the NetScaler command line:

At the command prompt, type:

- **show sourceroutecachetable**

Example:

```
> show sourceroutecachetable
SOURCEIP             MAC                  VLAN    INTERFACE
2001:DB8:5001:10     56:53:24:3d:02:eb     30          1/2
2001:DB8:5003:30     60:54:35:3e:04:bd     60          1/3
Done
```

Clearing the Source Route Cache Table

You can clear all the entries from the source route cache table on a NetScaler appliance.

To clear the source route cache table by using the NetScaler command line:

At the command prompt, type:

- **flush ns sourceroutecachetable**

Access Control Lists

September 21, 2020

Access Control Lists (ACLs) filter IP traffic and secure your network from unauthorized access. An ACL is a set of conditions that the NetScaler appliance evaluates to determine whether to allow access. For example, the Finance department probably does not want to allow its resources to be accessed by other departments, such as HR and Documentation, and those departments want to restrict access to their data.

When the NetScaler appliance receives a data packet, it compares the information in the data packet with the conditions specified in the ACL and allows or denies access. The administrator of the organization can configure ACLs to function in the following processing modes:

- ALLOW—Process the packet.
- BRIDGE—Bridge the packet to the destination without processing it. The packet is directly sent by Layer 2 and Layer 3 forwarding.
- DENY—Drop the packet.

ACL rules are the first level of defense on the NetScaler appliance.

NetScaler supports the following types of ACLs:

- **Simple ACLs** filter packets on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain. Any packet that has the characteristics specified in the ACL is dropped.

- **Extended ACLs** filter data packets on the basis of various parameters, such as source IP address, source port, action, and protocol. An extended ACL defines the conditions that a packet must satisfy for the NetScaler appliance to process the packet, bridge the packet, or drop the packet.

Nomenclature

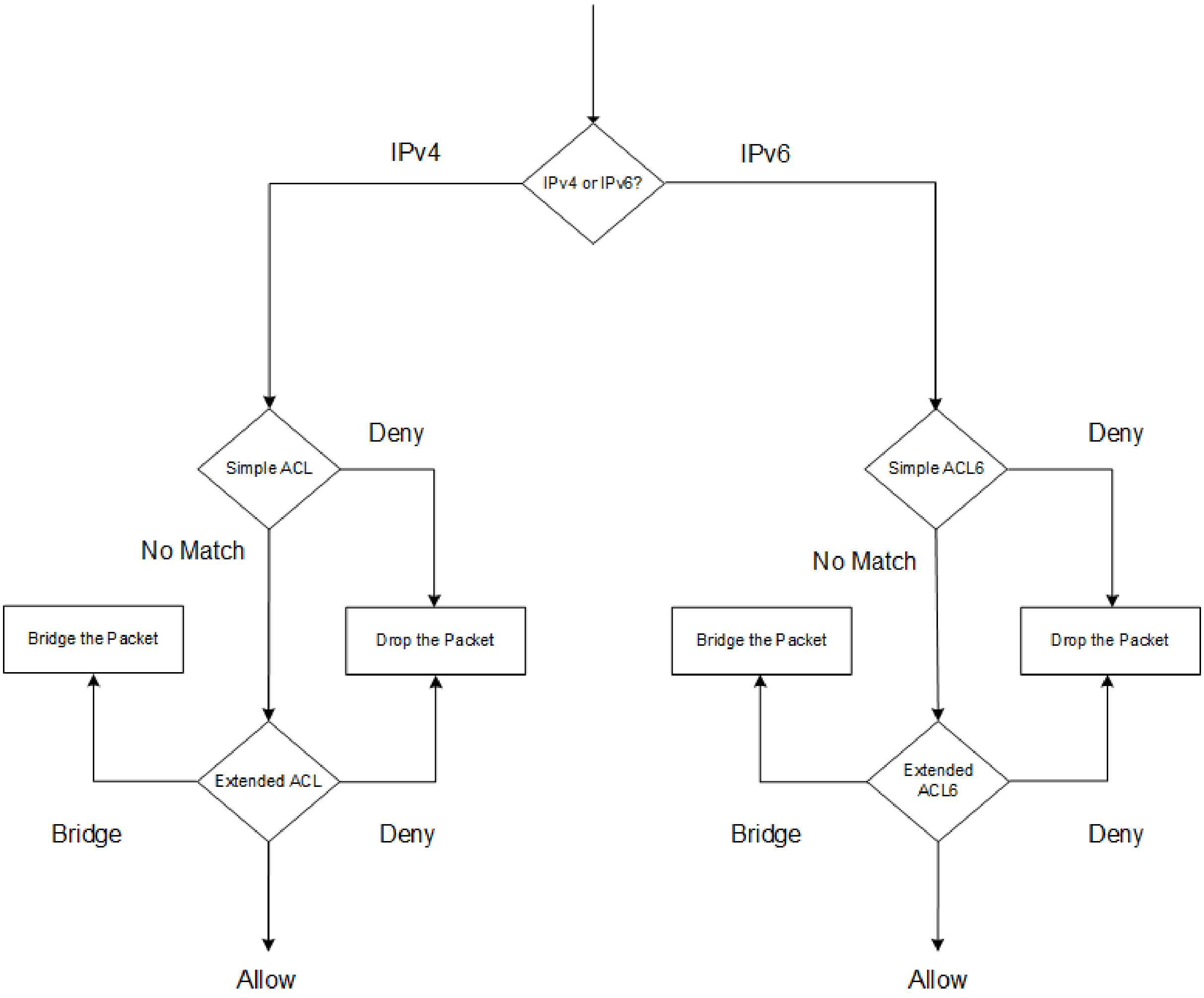
In the NetScaler user interfaces, the terms simple ACL and extended ACL refer to ACLs that process IPv4 packets. An ACL that processes IPv6 packets is called a simple ACL6 and or extended ACL6. When discussing both types, this documentation sometimes refers to both of them as simple ACLs or extended ACLs.

ACL Precedence

If both simple and extended ACLs are configured, incoming packets are compared to the simple ACLs first.

The NetScaler appliance first determines whether the incoming packet is an IPv4 or an IPv6 packet, and then compares the packet's characteristics to either simple ACLs or simple ACL6s. If a match is found, the packet is dropped. If no match is found, the packet is compared to extended ACLs or extended ACL6s. If that comparison results in a match, the packet is handled as specified in the ACL. The packet can be bridged, dropped, or allowed. If no match is found, the packet is allowed.

Figure 1. Simple and Extended ACLs Flow Sequence


IPv4
IPv4 or IPv6?
IPv6
Simple ACL
Deny
No Match
Bridge the Packet
Drop the Packet
Extended ACL
Bridge
Deny
Allow
Simple ACL6
Deny
No Match
Bridge the Packet
Drop the Packet
Extended ACL6
Bridge
Deny
Allow

Simple ACLs and Simple ACL6s

September 21, 2020

A simple ACL or simple ACL6 uses few parameters and can be configured only to drop IP packets. Packets can be dropped on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain.

When creating a simple ACL or simple ACL6, you can specify a time to live (TTL), in seconds, after which the ACL expires. ACLs with TTLs are not saved when you save the configuration. You can display simple ACLs and simple ACL6s to verify their configuration, and you can display their statistics.

Configuring Simple ACLs and Simple ACL6s

Configuring a simple ACL or simple ACL6 on a NetScaler appliance can include the following tasks.

- **Create simple ACLs or simple ACL6s**. Creating simple ACLs or simple ACL6s to drop (deny) packets on the basis of their source IP address and, optionally, their protocol, destination port, or traffic domain.
- **Remove simple ACLs or simple ACL6s**. These ACLs cannot be modified once created. If you need to modify a simple ACL or simple ACL6, you must remove it and create a new one.

NetScaler command line procedures

To create a simple ACL by using the NetScaler command line:

At the command prompt, type:

- **add ns simpleacl** <aclname> DENY -**srcIP** <ip_addr> [-**destPort**<port> -**protocol** (TCP | UDP)] [-**TTL** <positive_integer>]
- **show ns simpleacl** [<aclname>]

Example:

```
> add simpleacl rule1 DENY -srcIP 10.102.29.5 -TTL 600
Done
<!--NeedCopy-->
```

To create a simple ACL6 by using the NetScaler command line:

At the command prompt, type:

- **add ns simpleacl6** <aclname> DENY -**srcIPv6** <ipv6_addr|null> [-**destPort**<port> -**protocol** (TCP | UDP)] [-**TTL** <positive_integer>]
- **show ns simpleacl6** [<aclname>]

Example:

```
>  add ns simpleacl6 rule1 DENY  - srcIPv6 3ffe:192:168:215::82 -destPort 80 -Protocol TCP -TTL 9000
 Done
<!--NeedCopy-->
```

To remove a single simple ACL by using the NetScaler command line:

At the command prompt, type:

- **rm ns simpleacl** <aclname>
- **show ns simpleacl**

To remove a single simple ACL6 by using the NetScaler command line:

At the command prompt, type:

- **rm ns simpleacl6**<aclname>

- **show ns simpleacl6**

To remove all simple ACLs by using the NetScaler command line:

At the command prompt, type:

- **clear ns simpleacl**
- **show ns simpleacl**

To remove all simple ACL6s by using the NetScaler command line:

At the command prompt, type:

- **clear ns simpleacl6**
- **show ns simpleacl6**

NetScaler GUI procedures

To create a simple ACL by using the NetScaler GUI:

Navigate to System > Network > ACLs and, on the **Simple ACLs** tab, add a new simple ACL.

To create a simple ACL6 by using the NetScaler GUI:

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, add a new simple ACL6.

To remove a single simple ACL by using the NetScaler GUI:

Navigate to System > Network > ACLs and, on the **Simple ACLs** tab, delete the simple ACL.

To remove a single simple ACL6 by using the NetScaler GUI:

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, delete the simple ACL6.

To remove all simple ACLs by using the NetScaler GUI:

1. Navigate to System > Network > ACLs.
2. On the **Simple ACLs** tab, in the **Action** list, click **Clear**.

To remove all simple ACL6s by using the NetScaler GUI:

1. Navigate to System > Network > ACLs.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Clear**.

Displaying Simple ACL and Simple ACL6 Statistics

You can display the simple ACL (or simple ACL6) statistics, which include the number of hits, the number of misses, and the number of simple ACLs configured.

The following table describes statistics you can display for simple ACLs and simple ACL6s.

Statistics	Indicates
ACL hits	Packets matching an ACL
ACL misses	Packets not matching any ACL
ACL count	Number of ACLs configured

NetScaler command line procedures

To display simple ACL statistics by using the NetScaler command line:

At the command prompt, type:

- **stat ns simpleacl**

Example:

```
> stat ns simpleacl

SimpleACL Statistics

                                          Rate (/s)      Total
SimpleACL hits                                    0          0
SimpleACL misses                                  0      51872
SimpleACLs count                                 --          2
Done
<!--NeedCopy-->
```

To display simple ACL6 statistics by using the NetScaler command line:

At the command prompt, type:

- **stat ns simpleacl6**

NetScaler GUI procedures

To display simple ACL statistics by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Simple ACLs** tab, select the ACL and click **Statistics**.

To display simple ACL6 statistics by using the NetScaler GUI:

Navigate to System > Network > ACLs and, on the **Simple ACL6s** tab, select the simple ACL6 and click **Statistics**.

Terminating Established Connections

For a simple ACL or simple ACL6, the NetScaler appliance blocks any new connections that match the conditions specified in the ACL. Packets related to existing connections that were established before the ACL was created are not blocked. To terminate previously established connections that match an existing ACL, you can run a flush operation from the NetScaler command line or the NetScaler GUI.

Flush can be useful in the following cases:

- You receive a list of blacklisted IP addresses and want to completely block those IP addresses from accessing the NetScaler appliance. In this case, you create simple ACLs or simple ACL6s to block any new connections from these IP addresses, and then flush any existing connections associated with those addresses.
- You want to terminate a large number of connections from a particular network without taking the time to terminate them one by one.

Before you begin

- When you run flush, the NetScaler appliance searches through all of its established connections and terminates those that match conditions specified in any of the simple ACLs configured on the appliance.
- If you plan to create more than one simple ACL and flush existing connections that match any of them, you can minimize the effect on performance by first creating all of the simple ACLs and then running flush only once.

NetScaler command line procedures

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the NetScaler command line:

At the command prompt, type:

- **flush simpleacl -estSessions**

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the NetScaler command line:

At the command prompt, type:

- **flush simpleacl6 -estSessions**

NetScaler GUI procedures

To terminate all established IPv4 connections that match any of your configured simple ACLs by using the NetScaler GUI:

1. Navigate to System > Network > ACLs.
2. On the **Simple ACLs** tab, in the **Action** list, click **Flush**.

To terminate all established IPv6 connections that match any of your configured simple ACL6s by using the NetScaler GUI:

1. Navigate to System > Network > ACLs.
2. On the **Simple ACL6s** tab, in the **Action** list, click **Flush**.

Extended ACLs and Extended ACL6s

September 21, 2020

Extended ACLs and extended ACL6s provide parameters and actions not available with simple ACLs. You can filter data on the basis of parameters such as source IP address, source port, action, and protocol. You can specify tasks to allow a packet, deny a packet, or bridge a packet.

Extended ACLs and ACL6s can be modified after they are created, and you can renumber their priorities to specify the order in which they are evaluated.

Note: If you configure both simple and extended ACLs, simple ACLs take precedence over extended ACLs.

The following actions can be performed on extended ACLs and ACL6s: Modify, Apply, Disable, Enable, Remove, and Renumber (the priority). You can display extended ACLs and ACL6s to verify their configuration, and you can display their statistics.

You can configure the NetScaler appliance to log details for packets that match an extended ACL. However, you cannot log details of packets that match an ext

Applying Extended ACLs and Extended ACL6s: Unlike simple ACLs and ACL6s, extended ACLs and ACL6s created on the NetScaler appliance do not work until they are applied. Also, if you make any modifications to an extended ACL or ACL6, such as disabling the ACLs, changing a priority, or deleting the ACLs, you must reapply the extended ACLs or ACL6s. You must also reapply them after enabling logging. The procedure to apply extended ACLs or ACL6s reapplies all of them. For example, if you have applied extended ACL rules 1 through 10, and you then create and apply rule 11, the first 10 rules are applied afresh.

If a session has a DENY ACL related to it, that session is terminated when you apply the ACLs.

Extended ACLs and ACL6s are enabled by default. When they are applied, the NetScaler appliance starts comparing incoming packets against them. However, if you disable them, they are not used until you reenable them, even if they are reapplied.

Renumbering the priorities of Extended ACLs and Extended ACL6s: Priority numbers determine the order in which extended ACLs or ACL6s are matched against a packet. An ACL with a lower priority number has a higher priority. It is evaluated before ACLs with higher priority numbers (lower priorities), and the first ACL to match the packet determines the action applied to the packet.

When you create an extended ACL or ACL6, the NetScaler appliance automatically assigns it a priority number that is a multiple of 10, unless you specify otherwise. For example, if two extended ACLs have priorities of 20 and 30, respectively, and you want a third ACL to have a value between those numbers, you might assign it a value of 25. If you later want to retain the order in which the ACLs are evaluated but restore their numbering to multiples of 10, you can use the renumber procedure.

Configuring Extended ACLs and Extended ACL6s

Configuring an extended ACL or ACL6 on a NetScaler appliance consists of the following tasks.

- **Create an extended ACL or ACL6.** Create an extended ACL or ACL6 to either allow, deny, or bridge a packet. You can specify an IP address or range of IP addresses to match against the source or destination IP addresses of the packets. You can specify a protocol to match against the protocol of incoming packets.
- (Optional) **Modify an extended ACL or ACL6**. You can modify extended ACLs or ACL6s that you previously created. Or, if you want to temporarily take one out of use you can disable it, and later reenable it.
- **Apply extended ACLs or ACL6s**. After you create, modify, disable or reenable, or delete an extended ACL or ACL6, you must apply the extended ACLs or ACL6s to activate them.
- (Optional) **Renumber the priorities of extended ACLs or ACL6s**. If you have configured ACLs with priorities that are not multiples of 10 and want to restore the numbering to multiples of 10, use the renumber procedure.

NetScaler command line procedures

To create an extended ACL by using the NetScaler command line:

At the command prompt, type:

- **add ns acl** <aclname> <aclaction> [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>]

[-**icmpType** <positive_integer> [-**icmpCode** <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)] [-**logstate** (ENABLED | DISABLED) [-**ratelimit** <positive_integer>]]
- **show ns acl** [<aclName>]

Example:

```
> add ns acl restrict DENY -srcport 45-1024 -destIP 192.168.1.1 -
   protocol TCP
 Done
<!--NeedCopy-->
```

To create an extended ACL6 by using the NetScaler command line:

At the command prompt, type:

- **add ns acl6** <acl6name> <acl6action> [-**srcIPv6** [<operator>] <srcIPv6Val>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIPv6** [<operator>] <destIPv6Val>] [-**destPort** [<operator>] <destPortVal>] [-**TTL** <positive_integer>] [-**srcMac** <mac_addr>] [(-**protocol** <protocol> [-established]) | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer>] [-**interface** <interface_name>] [-**icmpType** <positive_integer> [-icmpCode <positive_integer>]] [-**priority** <positive_integer>] [-**state** (ENABLED | DISABLED)]
- **show ns acl6** [<aclName>]

Example:

```
> add ns acl6 rule6  DENY -srcport 45-1024 -destIPv6 2001::45 -protocol
    TCP
 Done
<!--NeedCopy-->
```

To modify an extended ACL by using the NetScaler command line:

To modify an extended ACL, type the **set ns acl** command, the name of the extended ACL, and the parameters to be changed, with their new values.

To modify an extended ACL6 by using the NetScaler command line:

To modify an extended ACL6, type the **set ns acl6** command, the name of the extended ACL6, and the parameters to be changed, with their new values.

To disable or enable an extended ACL by using the NetScaler command line:

At the command prompt, type one of the following commands:

- **disable ns acl** <aclname>
- **enable ns acl** <aclname>

To disable or enable an extended ACL6 by using the NetScaler command line:

At the command prompt, type one of the following commands:

- **disable ns acl6** <aclname>
- **enable ns acl6**<aclname>

To apply extended ACLs by using the NetScaler command line:

At the command prompt, type:

- **apply ns acls**

To apply extended ACL6s by using the NetScaler command line:

At the command prompt, type:

- **apply ns acls6**

To renumber the priorities of extended ACLs by using the NetScaler command line:

At the command prompt, type:

- **renumber ns acls**

To renumber the priorities of extended ACL6s by using the NetScaler command line:

At the command prompt, type:

- **renumber ns acls6**

NetScaler GUI procedures

To configure an extended ACL by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, add a new extended ACL or edit an existing extended ACL. To enable or disable an existing extended ACL, select it, and then select **Enable** or **Disable** from the **Action** list.

To configure an extended ACL6s by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, add a new extended ACL6 or edit an existing extended ACL6. To enable or disable an existing extended ACL6, select it, and then select **Enable** or **Disable** from the **Action** list.

To apply extended ACLs by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Apply**.

To apply extended ACL6s by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Apply**.

To renumber the priorities of extended ACLs by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab, in the **Action** list, click **Renumber Priority (s)**.

To renumber the priorities of extended ACL6s by using the NetScaler GUI:

Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab, in the **Action** list, click **Renumber Priority (s)**.

Sample Configurations

The following table shows examples of configuring extended ACL rules through the command line interface: ACLs sample configurations.

Logging Extended ACLs

You can configure the NetScaler appliance to log details for packets that match extended ACLs.

Note: You cannot enable logging for extended ACL6s.

In addition to the ACL name, the logged details include packet-specific information such as the source and destination IP addresses. The information is stored either in the syslog file or in the nslog file, depending on the type of global logging (syslog or nslog) enabled.

Logging must be enabled at both the global level and the ACL level. The global setting takes precedence. For more information about enabling logging globally, see "."

To optimize logging, when multiple packets from the same flow match an ACL, only the first packet's details are logged, and the counter is incremented for every packet that belongs to the same flow. A flow is defined as a set of packets that have the same values for the source IP address, destination IP address, source port, destination port, and protocol parameters. To avoid flooding of log messages, the NetScaler appliance performs internal rate limiting so that packets belonging to the same flow are not repeatedly logged. The total number of different flows that can be logged at any given time is limited to 10,000.

Note: You must apply ACLs after you enable logging.

NetScaler command line procedures

To configure extended ACL Logging by using the NetScaler command line:

At the command prompt, type the following commands to configure logging and verify the configuration:

- **set ns acl** <aclName> [-**logState** (ENABLED | DISABLED)] [-**rateLimit** <positive_integer>]
- **show ns acl** [<aclName>]

Example:

```
> set ns acl restrict -logstate ENABLED -ratelimit 120
Warning: ACL modified, apply ACLs to activate change
<!--NeedCopy-->
```

NetScaler GUI procedures

To configure extended ACL Logging by using the NetScaler GUI:

1. Navigate to System > Network > ACLs and, on the **Extended ACLs** tab, open the extended ACL.
2. Set the following parameters:
    - **Log State**—Enable or disable logging of events related to the extended ACL rule. The log messages are stored in the configured syslog or auditlog server.
    - **Log Rate Limit**—Maximum number of log messages to be generated per second. If you set this parameter, you must enable the Log State parameter.

Displaying Extended ACL and Extended ACL6s Statistics

You can display statistics of extended ACLs and ACL6s.

The following table lists the statistics associated with extended ACLs and ACL6s, and their descriptions.

Statistic	Specifies
Allow ACL hits	Packets matching ACLs with processing mode set to ALLOW. The NetScaler appliance processes these packets.
NAT ACL hits	Packets matching a NAT ACL, resulting in a NAT session.
Deny ACL hits	Packets dropped because they match ACLs with processing mode set to DENY.
Bridge ACL hits	Packets matching a bridge ACL, which in transparent mode bypasses service processing.
ACL hits	Packets matching an ACL.
ACL misses	Packets not matching any ACL.

NetScaler command line procedures

To display the statistics of all extended ACLs by using the NetScaler command line:

At the command prompt, type:

- **stat ns acl**

To display the statistics of all extended ACL6s by using the NetScaler command line:

At the command prompt, type:

- **stat ns acl6**

NetScaler GUI procedures

To display the statistics of an extended ACL by using the NetScaler GUI:

Navigate to System > Network > ACLs, on the **Extended ACLs** tab, select the extended ACL, and click **Statistics**.

To display the statistics of an extended ACL6 by using the NetScaler GUI:

Navigate to System > Network > ACLs, on the **Extended ACL6s** tab, select the extended ACL, and click **Statistics**.

Stateful ACLs

A stateful ACL rule creates a session when a request matches the rule and allows the resulting responses even if these responses matches a deny ACL rule in NetScaler appliance. A stateful ACL offloads the work of creating additional ACL rules/forwarding session rules for allowing these specific responses.

Stateful ACLs can be best used in an edge firewall deployment of a NetScaler appliance having the following requirements:

- The NetScaler appliance must allow requests initiated from internal clients and the related responses from the Internet.
- The appliance must drop the packets from the Internet that are not related to any client connections.

Before you begin

Before you configure stateful ACL rules, note the following points:

- The NetScaler appliance supports stateful ACL rules as well as stateful ACL6 rules.
- In a high availability setup, the sessions for a stateful ACL rule are not synchronized to the secondary node.

- You cannot configure an ACL rule as stateful if the rule is bound to any NetScaler NAT configuration. Some examples of NetScaler NAT configurations are:
  - RNAT
  - Large Scale NAT (large scale NAT44, DS-Lite, large scale NAT64)
  - NAT64
  - Forwarding session
- You cannot configure an ACL rule as stateful if TTL and Established parameters are set for this ACL rule.
- The sessions created for a stateful ACL rule continue to exist till timeout irrespective of the following ACL operations:
  - Remove ACL
  - Disable ACL
  - Clear ACL
- Stateful ACLs are not supported for the following protocols:
  - Active FTP
  - TFTP

Configure stateful IPv4 ACL rules

Configuring a stateful ACL rule consists of enabling the stateful parameter of an ACL rule.

To enable the stateful parameter of an ACL rule by using the NetScaler command line:

- To enable the stateful parameter while adding an ACL rule, at the command prompt, type:
  - add acl <lname> ALLOW -stateful (**ENABLED** | **DISABLED**)
  - apply acls
  - show acl <name>
- To enable the stateful parameter of an existing ACL rule, at the command prompt, type:
  - set acl <name> -**stateful** (**ENABLED** | **DISABLED**)
  - apply acls
  - show acl <name>

To enable the stateful parameter of an ACL rule by using the NetScaler command line:

1. Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACLs** tab.
2. Enable the **Stateful** parameter while adding, or modifying an existing ACL rule.

```
> add acl ACL-1 allow -srciP 1.1.1.1 -stateful Yes

Done

> apply acls

```

```
Done

> show acl

1)          Name: ACL-1

    Action: ALLOW                               Hits: 0

    srcIP = 1.1.1.1

    destIP

    srcMac:

    Protocol:

    Vlan:                                      Interface:

    Active Status: ENABLED                      Applied Status: NOTAPPLIED

    Priority: 10                                NAT: NO

    TTL:

    Log Status: DISABLED

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

Configure stateful ACL6 rules

Configuring a stateful ACL6 rule consists of enabling the stateful parameter of an ACL6 rule.

To enable the stateful parameter of an ACL6 rule by using the NetScaler command line:

- To enable the stateful parameter while adding an ACL6 rule, at the command prompt, type:
  - add acl6 <name> ALLOW -**stateful** (**ENABLED** | **DISABLD**)
  - apply acls6
  - show acl6 <name>
- To enable the stateful parameter of an existing ACL6 rule, at the command prompt, type:
  - set acl6 <name> -**stateful** (**ENABLED** | **DISABLED**)

  - apply acls6
  - show acl6 <name>

To enable the stateful parameter of an ACL6 rule by using the NetScaler command line:

1. Navigate to **System** > **Network** > **ACLs** and, on the **Extended ACL6s** tab.
2. Enable the **Stateful** parameter while adding, or modifying an existing ACL6 rule.

```
>  add acl6 ACL6-1 allow -srcipv6 1000::1  - stateful Yes

Done

>  apply acls6

Done

> show acl6

1)    Name: ACL6-1

    Action: ALLOW                          Hits: 0

    srcIPv6 = 1000::1

    destIPv6

    srcMac:

    Protocol:

    Vlan:                                 Interface:

    Active Status: ENABLED                 Applied Status: NOTAPPLIED

    Priority: 10                           NAT: NO

    TTL:

    Forward Session: NO

    Stateful: YES
<!--NeedCopy-->
```

MAC Address Wildcard Mask for ACLs

September 21, 2020

A wildcard mask parameter has been introduced for extended ACLs and ACL6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of incoming packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask should meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an ACL, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the incoming packets.

To specify a range of source MAC addresses in an ACL rule by using the NetScaler command line:

At the command prompt, type:

- **add ns acl** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns acl** <aclname>

Example:

```
> add ns acl ACL-1 ALLOW -protocol TCP -srcport 2000-3000 -srcMac 96:fa:95:1d:67:4a
  -srcMacMask 000000111111
```

```
Done
<!--NeedCopy-->
```

To specify a range of source MAC addresses in an ACL6 rule by using the NetScaler command line:

At the command prompt, type:

- **add ns acl6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns acl6** <acl6name>

Example:

```
> add ns acl6 ACL6-1 ALLOW -destIPv6 2001::45 -srcMac 96:fa:90:1d:67:4a
 - srcMacMask 000000001111
Done
<!--NeedCopy-->
```

Blocking Traffic on Internal Ports

September 21, 2020

The NetScaler appliance does not block traffic that matches an ACL rule if the traffic is destined to the appliance's NSIP address, or one of its SNIP addresses, and a port in the 3008-3011 range.

This behavior is now specified by the default setting of the new Implicit ACL Allow (implicitACLAllow) parameter (of the L3 param command). You can disable this parameter if you want to block traffic to ports in the 3008-3011 range. An appliance in a high availability configuration makes an exception for its partner (primary or secondary) node. It does not block traffic from that node.

To disable or enable this parameter by using the NetScaler command line:

At the command prompt, type:

set l3param -**implicitACLAllow** [ENABLED	DISABLED]

-
- **sh l3param**

Note: The parameter implicitACLAllow is enabled by default.

Example:

```
> set l3param -implicitACLAllow DISABLED
Done
```

```
<!--NeedCopy-->
```

IP Routing

September 21, 2020

NetScaler appliances support both dynamic and static routing. Because simple routing is not the primary role of a NetScaler, the main objective of running dynamic routing protocols is to enable route health injection (RHI), so that an upstream router can choose the best among multiple routes to a topographically distributed virtual server.

Most NetScaler implementations use some static routes to reduce routing overhead. You can create backup static routes and monitor routes to enable automatic switchover in the event that a static route goes down. You can also assign weights to facilitate load balancing among static routes, create null routes to prevent routing loops, and configure IPv6 static routes. You can configure policy based routes (PBRs), for which routing decisions are based on criteria that you specify.

Configuring Dynamic Routes

September 21, 2020

When a dynamic routing protocol is enabled, the corresponding routing process monitors route updates and advertises routes. Routing protocols enable an upstream router to use the equal cost multipath (ECMP) technique to load balance traffic to identical virtual servers hosted on two standalone NetScaler appliances. Dynamic routing on a NetScaler appliance uses three routing tables. In a high-availability setup, the routing tables on the secondary appliance mirror those on the primary.

For command reference guides and unsupported commands on dynamic routing protocol, see Dynamic Routing Protocol Command Reference Guides and Unsupported Commands.

The NetScaler supports the following protocols:

- Routing Information Protocol (RIP) version 2
- Open Shortest Path First (OSPF) version 2
- Border Gateway Protocol (BGP)
- Routing Information Protocol next generation (RIPng) for IPv6
- Open Shortest Path First (OSPF) version 3 for IPv6
- ISIS Protocol

You can enable more than one protocol simultaneously.

Routing Tables in NetScaler

In a NetScaler appliance, the NetScaler kernel routing table, the FreeBSD kernel routing table, and the NSM FIB routing table each hold a different set of routes and serve a different purpose. They communicate with each other by using UNIX routing sockets. Route updates are not automatically propagated from one routing table to another. You must configure propagation of route updates for each routing table.

NS Kernel Routing Table

The NS kernel routing table holds subnet routes corresponding to the NSIP and to each SNIP and MIP. Usually, no routes corresponding to VIPs are present in the NS kernel routing table. The exception is a VIP added by using the add ns ip command and configured with a subnet mask other than 255.255.255.255. If there are multiple IP addresses belonging to the same subnet, they are abstracted as a single subnet route. In addition, this table holds a route to the loopback network (127.0.0.0) and any static routes added through the NetScaler command line (CLI). The entries in this table are used by the NetScaler in packet forwarding. From the NetScaler CLI, they can be inspected with the show route command.

FreeBSD Routing Table

The sole purpose of the FreeBSD routing table is to facilitate initiation and termination of management traffic (telnet, ssh, etc.). In a NetScaler appliance, these applications are tightly coupled to FreeBSD, and it is imperative for FreeBSD to have the necessary information to handle traffic to and from these applications. This routing table contains a route to the NSIP subnet and a default route. In addition, FreeBSD adds routes of type WasCloned(W) when the NetScaler establishes connections to hosts on local networks. Because of the highly specialized utility of the entries in this routing table, all other route updates from the NS kernel and NSM FIB routing tables bypass the FreeBSD routing table. Do not modify it with the route command. The FreeBSD routing table can be inspected by using the netstat command from any UNIX shell.

Network Services Module (NSM) FIB

The NSM FIB routing table contains the advertisable routes that are distributed by the dynamic routing protocols to their peers in the network. It may contain:

- **Connected routes**. IP subnets that are directly reachable from the NetScaler. Typically, routes corresponding to the NSIP subnet and subnets over which routing protocols are enabled are present in NSM FIB as connected routes.
- **Kernel routes**. All the VIP addresses on which the -hostRoute option is enabled are present in NSM FIB as kernel routes if they satisfy the required RHI Levels. In addition, NSM FIB contains

any static routes configured on the NetScaler CLI that have the - advertise option enabled. Alternatively, if the NetScaler is operating in Static Route Advertisement (SRADV) mode, all static routes configured on the NetScaler CLI are present in NSM FIB. These static routes are marked as kernel routes in NSM FIB, because they actually belong to the NS kernel routing table.
- **Static routes**. Normally, any static route configured in VTYSH is present in NSM FIB. If administrative distances of protocols are modified, this may not always be the case. An important point to note is that these routes can never get into the NS kernel routing table.
- **Learned routes**. If the NetScaler is configured to learn routes dynamically, the NSM FIB contains routes learned by the various dynamic routing protocols. Routes learned by OSPF, however, need special processing. They are downloaded to FIB only if the fib-install option is enabled for the OSPF process. This can be done from the router-config view in VTYSH.

Dynamic Routing in a High Availability Setup

In a high availability setup, the primary node runs the routing process and propagates routing table updates to the secondary node. The routing table of the secondary node mirrors the routing table on the primary node.

Non-Stop Forwarding

After failover, the secondary node takes some time to start the protocol, learn the routes, and update its routing table. But this does not affect routing, because the routing table on the secondary node is identical to the routing table on the primary node. This mode of operation is known as non-stop forwarding.

Black Hole Avoidance Mechanism

After failover, the new primary node injects all its VIP routes into the upstream router. However, that router retains the old primary node’s routes for 180 seconds. Because the router is not aware of the failover, it attempts to load balance traffic between the two nodes. During the 180 seconds before the old routes expire, the router sends half the traffic to the old, inactive primary node, which is, in effect, a black hole.

To prevent this, the new primary node, when injecting a route, assigns it a metric that is slightly lower than the one specified by the old primary node.

Interfaces for Configuring Dynamic Routing

To configure dynamic routing, you can use either the NetScaler GUI or a command-line interface. The NetScaler supports two independent command-line interfaces: the NetScaler CLI and the Virtual Teletype Shell (VTYSH). The NetScaler CLI is the appliance's native shell. VTYSH is exposed by ZebOS. The NetScaler routing suite is based on ZebOS, the commercial version of GNU Zebra.

> **Note:**
>
> Citrix recommends that you use VTYSH for all commands except those that can be configured only on the NetScaler CLI. Use of the NetScaler CLI should generally be limited to commands for enabling the routing protocols, configuring host route advertisement, and adding static routes for packet forwarding.

Dynamic Routing Protocol Command Reference Guides and Unsupported Commands

The following table lists command reference guide links, for various dynamic routing protocols, and unsupported commands on the NetScaler appliance: Dynamic routing protocol reference guides and unsupported commands.

Configuring RIP

September 21, 2020

Routing Information Protocol (RIP) is a Distance Vector protocol. The NetScaler supports RIP as defined in RFC 1058 and RFC 2453. RIP can run on any subnet.

After enabling RIP, you need to configure advertisement of RIP routes. For troubleshooting, you can limit RIP propagation. You can display RIP settings to verify the configuration.

Enabling and Disabling RIP

Use either of the following procedures to enable or disable RIP. After you enable RIP, the NetScaler appliance starts the RIP process. After you disable RIP, the appliance stops the RIP process.

To enable or disable RIP routing by using the NetScaler command line:

At the command prompt, enter one of the following commands to enable or disable RIP:

- **enable ns feature RIP**
- **disable ns feature RIP**

To enable or disable RIP routing by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change advanced features**.
2. Select or clear the **RIP Routing** option.

Advertising Routes

RIP enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure RIP to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router rip	Start the RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting RIP Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given interface.

To limit RIP propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.

Command	Specifies
router rip	Start the RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the RIP Configuration

You can display the routing table and other RIP settings.

To view the RIP settings by using the VTYSH command line:

At the command prompt, type the following commands in the following order:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh rip	Display updated RIP routing table.
sh rip interface <vlan_name>	Displays RIP information for the specified VLAN.

Example:

```
NS# VTYSH
NS# sh rip
NS# sh rip interface VLAN0
<!--NeedCopy-->
```

Configuring OSPF

September 21, 2020

The NetScaler supports Open Shortest Path First (OSPF) Version 2 (RFC 2328). The features of OSPF on the NetScaler are:

- If a vserver is active, the host routes to the vserver can be injected into the routing protocols.
- OSPF can run on any subnet.
- Route learning advertised by neighboring OSPF routers can be disabled on the NetScaler.
- The NetScaler can advertise Type-1 or Type-2 external metrics for all routes.
- The NetScaler can advertise user-specified metric settings for VIP routes. For example, you can configure a metric per VIP without special route maps.
- You can specify the OSPF area ID for the NetScaler.
- The NetScaler supports not-so-stubby-areas (NSSAs). An NSSA is similar to an OSPF stub area but allows injection of external routes in a limited fashion into the stub area. To support NSSAs, a new option bit (the N bit) and a new type (Type 7) of Link State Advertisement (LSA) area have been defined. Type 7 LSAs support external route information within an NSSA. An NSSA area border router (ABR) translates a type 7 LSA into a type 5 LSA that is propagated into the OSPF domain. The OSPF specification defines only the following general classes of area configuration:
  - Type 5 LSA: Originated by routers internal to the area are flooded into the domain by AS boarder routers (ASBRs).
  - Stub: Allows no type 5 LSAs to be propagated into/throughout the area and instead depends on default routing to external destinations.

After enabling OSPF, you need to configure advertisement of OSPF routes. For troubleshooting, you can limit OSPF propagation. You can display OSPF settings to verify the configuration.

Enabling and Disabling OSPF

To enable or disable OSPF, you must use either the NetScaler command line or the NetScaler GUI. When OSPF is enabled, the NetScaler starts the OSPF process. When OSPF is disabled, the NetScaler stops the OSPF routing process.

To enable or disable OSPF routing by using the NetScaler command line:

At the command prompt, type one of the following commands:

1. **enable ns feature OSPF**
2. **disable ns feature OSPF**

To enable or disable OSPF routing by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in **Modes and Features** group, click **Change advanced features**.
2. Select or clear the **OSPF Routing** option.

Advertising OSPF Routes

OSPF enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure OSPF to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enters global configuration mode.
router OSPF	Start OSPF routing process and enter configuration mode for the routing process.
network A.B.C.D/M area <0-4294967295>	Enable routing on an IP network.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# network 10.102.29.0/24 area 0
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting OSPF Propagations

If you need to troubleshoot your configuration, you can configure listen-only mode on any given VLAN.

To limit OSPF propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router OSPF	Start OSPF routing process and enters configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the OSPF Configuration

You can display current OSPF neighbors, and OSPF routes.

To view the OSPF settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh OSPF neighbor	Displays current neighbors.
sh OSPF route	Displays OSPF routes.

Example:

```
>VTYSH
NS# sh ip OSPF neighbor
NS# sh ip OSPF route
<!--NeedCopy-->
```

Configuring Graceful Restart for OSPF

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

To configure graceful restart for OSPF by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id <id>	NS(config)# router-id 1.1.1.1	Sets a router identifier for the NetScaler appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability set up for graceful restart to work properly in the HA setup.
ospf restart grace-period <1-1800>	NS(config)# ospf restart grace-period 170	Specifies the grace period, in seconds, for which the routes are to be preserved in the helper devices. Default value: 120 seconds.

Command	Example	Command Description
ospf restart helper max-grace-period <1-1800>	NS(config)# ospf restart helper max-grace-period 180	This is an optional command to limit the maximum grace period for which the NetScaler appliance will be in the helper mode. If the NetScaler appliance receives an opaque LSA with grace-period greater than the set helper max-grace-period, the LSA is discarded and the NetScaler is not placed in helper mode.
router ospf	NS(config)# router ospf	Starts OSPF routing process and enter configuration mode for the routing process.
network A.B.C.D/M area <0-4294967295>	NS(config-router)# network 192.0.2.0/24 area 0	Enables routing on an IP network.
capability restart graceful	NS(config-router)# capability restart graceful	Enables graceful restart on the OSPF routing process.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring BGP

September 21, 2020

The NetScaler appliance supports BGP (RFC 4271). The features of BGP on the NetScaler are:

- The NetScaler advertises routes to BGP peers.
- The NetScaler injects host routes to virtual IP addresses (VIPs), as determined by the health of the underlying virtual servers.
- The NetScaler generates configuration files for running BGP on the secondary node after failover in an HA configuration.
- This protocol supports IPv6 route exchanges.
- As-Override Support in Border Gateway Protocol

After enabling BGP, you need to configure advertisement of BGP routes. For troubleshooting, you can limit BGP propagation. You can display BGP settings to verify the configuration.

Enabling and Disabling BGP

To enable or disable BGP, you must use either the NetScaler command line or the NetScaler GUI. When BGP is enabled, the NetScaler appliance starts the BGP process. When BGP is disabled, the appliance stops the BGP process.

To enable or disable BGP routing by using the NetScaler command line:

At the command prompt, type one of the following commands:

- enable ns feature BGP
- disable ns feature BGP

To enable or disable BGP routing by using the NetScaler GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the BGP Routing option.

Advertising IPv4 Routes

You can configure the NetScaler appliance to advertise host routes to VIPs and to advertise routes to downstream networks.

To configure BGP to advertise IPv4 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the link local IPv4 address of the neighbor in the specified autonomous system.
Address-family ipv4	Enter address family configuration mode.
Neighbor < IPv4 address> activate	Exchange prefixes for the IPv4 router family between the peer and the local node by using the link local address.

Command	Specifies
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor 10.102.29.170 remote-as 100
NS(config-router)# Address-family ipv4
NS(config-router-af)# Neighbor 10.102.29.170 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

Prerequisites for IPv6 BGP

Before you begin configuring IPv6 BGP, do the following:

- Make sure that you understand the IPv6 BGP protocol.
- Enable the IPv6 feature.

Advertising IPv6 BGP Routes

Border Gateway Protocol (BGP) enables an upstream router to load balance traffic between two identical virtual servers hosted on two standalone NetScaler appliances. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure BGP to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter. Possible values: 1 to 4,294,967,295.

Command	Specifies
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv6 BGP neighbor table with the link local IPv6 address of the neighbor in the specified autonomous system.
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the peer and the local node by using the link local address.
redistribute kernel	Redistribute kernel routes.
redistribute static	Redistribute static routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router BGP 5
NS(config-router)# Neighbor a1bc::102 remote-as 100
NS(config-router)# Address-family ipv6
NS(config-router-af)# Neighbor a1bc::102 activate
NS(config-router)# redistribute kernel
NS(config-router)# redistribute static
<!--NeedCopy-->
```

Verifying the BGP Configuration

You can use VTYSH to display BGP settings.

To view the BGP settings using the VTYSH command line

At the command prompt, type:

```
VTYSH
You are now in the VTYSH command prompt. An output similar to the following appears:
NS170#
At the VTYSH command prompt, type:
NS170# sh ip BGP
NS170# sh BGP
NS170# sh ip BGP neighbors
NS170# sh ip BGP summary
NS170# sh ip BGP route-map <map-tag>
```

```
<!--NeedCopy-->
```

As-Override Support in Border Gateway Protocol

As a part of BGP loop prevention functionality, if a router receives a BGP packet containing the router's Autonomous System Number (ASN) in the Autonomous Systems (AS) path, the router drops the packet. The assumption is that the packet originated from the router and has reached the place from where it originated.

If an enterprise has several sites with a same ASN, BGP loop prevention causes the sites with an identical ASN to not get linked by another ASN. Routing updates (BGP packets) are dropped when another site receives them.

To solve this issue, BGP AS-Override functionality has been added to the ZebOS BGP routing module of the NetScaler.

With AS-Override enabled for a peer device, when the NetScaler appliance receives a BGP packet for forwarding to the peer, and the ASN of the packet matches that of the peer, the appliance replaces the ASN of the BGP packet with its own ASN number before forwarding the packet.

You can enable AS-Override for a specific neighbor or a group of neighbors (peer group) by using the VTYSH command line.

To configure BGP AS-Override for a IPv4 neighbor by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv4 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv4 address> as-override	Enable BGP as-override for the specified neighbor.

```
    > VTYSH NS# configure terminal
    NS(config)# router BGP 5
    NS(config-router)# Neighbor 192.0.2.100 remote-as 100
    NS(config-router)# Neighbor 10.102.29.100 as-override
<!--NeedCopy-->
```

To configure BGP AS-Override for a IPv4 BGP peer group by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbot <IPv4 address> **peer-group** <peer group name>	Associate neighbors to the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

```
    > VTYSH NS# configure terminal
    NS(config)# router bgp 5
    NS(config-router)# neighbor external-peers-1 peer-group
    NS(config-router)# neighbor 192.0.2.101 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.102 peer-group external-peers-1
    NS(config-router)# neighbor 192.0.2.103 peer-group external-peers-1
    NS(config-router)# Neighbor external-peers-1 remote-as 100
    NS(config-router)# Neighbor external-peers-1 as-override
<!--NeedCopy-->
```

To configure BGP AS-Override for an IPv6 neighbor by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor < IPv6 address> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.

Command	Specifies
Address-family ipv6	Enter address family configuration mode.
Neighbor < IPv6 address> activate	Exchange prefixes for the IPv6 router family between the specified neighbor and the NetScaler by using the link local address.
Neighbor <IPv6 address> as-override	Enable BGP as-override for the specified neighbor.

```
    > VTYSH NS# configure terminal
    NS(config)# router bgp 5
    NS(config-router)# Neighbor a1bc::102 remote-as 100
    NS(config-router)# Neighbor a1bc::102 as-override
    NS(config-router)# Address-family ipv6
    NS(config-router-af)# Neighbor a1bc::102 activate
    NS(config-router)# Neighbor a1bc::102 as-override
<!--NeedCopy-->
```

To configure BGP AS-Override for IPv6 peer group by using the VTYSH command line:

Command	Specifies
configure terminal	Enter global configuration mode.
router BGP < ASnumber>	BGP autonomous system. < ASnumber> is a required parameter.
Neighbor <peer group name> **peer-group**	Create a BGP peer group.
Neighbor <IPv6 address> **peer-group** <peer group name>	Associate a neighbor with the specified peer group.
Neighbor <peer group name> remote-as < as-number>	Update the IPv4 BGP neighbor table with the IPv4 address of the neighbor in the specified autonomous system.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.
Address-family ipv6	Enter address family configuration mode.

Command	Specifies
Neighbor <peer group name> activate	Exchange prefixes for the IPv6 router family between the neighbors of the specified peer group and the NetScaler by using the link local address.
Neighbor <peer group name> as-override	Enable BGP as-override for all the neighbors that are associated with the specified peer group.

```
      > VTYSH NS# configure terminal
      NS(config)# router bgp 5
      NS(config-router)# neighbor external-peers-2 peer-group
      NS(config-router)# neighbor 2001::1 peer-group external-peers-2
      NS(config-router)# neighbor 2001::2 peer-group external-peers-2
      NS(config-router)# Neighbor external-peers-2 remote-as 100
      NS(config-router)# Neighbor external-peers-2 as-override
      NS(config-router)# Address-family ipv6
      NS(config-router-af)# Neighbor external-peers-2 activate
      NS(config-router)# Neighbor external-peers-2 as-override
<!--NeedCopy-->
```

Graceful Restart

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped. Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

Configuring Graceful Restart for BGP

To configure graceful restart for BGP by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id <ID>	NS(config)# router-id 1.1.1.1	A router identifier for the NetScaler appliance. This identifier is set for all the dynamic routing protocols. The same identifier must be specified on the other node in a high availability setup for graceful restart to work properly.
router bgp <AS-number>	NS(config)# router bgp 5	Enters BGP configuration mode.
bgp graceful-restart	NS(config)# bgp graceful-restart	Enables graceful restart on the BGP routing process.
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	Specifies the grace period, in seconds, that the helper routers waits for a TCP connection from the new primary node after a failover. For this amount of time, the helper routers preserve the routes.
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	Specifies the time, in seconds, that the NetScaler appliance in helper mode retains the stale routes for restarting neighbor routers. The default value is 360 seconds.
neighbor <IPv4 address of the peer router> remote-as <AS-number>	NS(config-router)# neighbor 192.0.2.30 remote-as 2	Establishes BGP peering with the specified neighbor router device.

Command	Example	Command Description
neighbor <IPv4 address of the peer router> capability graceful-restart	NS(config-router)# neighbor 192.0.2.30 capability graceful-restart	Enables graceful restart with the specified neighbor.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring Graceful Restart for IPv6 BGP

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

To configure graceful restart for IPv6 BGP by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id <id>	NS(config)# router-id 1.1.1.1	Sets a router identifier for the NetScaler appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability setup for graceful restart to work properly.
router bgp <AS-number>	NS(config)# router bgp 5	Enters configuration mode for BGP protocol.

Command	Example	Command Description
bgp graceful-restart	NS(config)# bgp graceful-restart	Enables graceful restart on the BGP routing process.
bgp graceful-restart restart-time <1-1800>	NS(config-router)# bgp graceful-restart restart-time 170	Specifies the grace period, in seconds, that the helper routers waits for a TCP connection from the new primary node after a failover. For this amount of time, the helper routers preserve the routes. The default value is 360 seconds.
bgp graceful-restart stalepath-time <1-1800>	NS(config-router)# bgp graceful-restart stalepath-time 180	Specifies the time, in seconds, that the NetScaler appliance in helper mode retains the stale routes for restarting neighbor routers. The default value is 360 seconds.
neighbor <IPv6 address> remote-as <AS-number>	NS(config-router)# neighbor 2001:db8::10 remote-as 2	Establishes BGP peering with the specified neighbor router device.
address-family ipv6	NS(config-router)#address-family ipv6	Enters address family configuration mode.
neighbor <IPv6 address of the neighbor> activate	NS(config-router-af)#neighbor 2001:db8::10 activate	Enables the exchange of address family routes with the specified neighbor router device.
neighbor <IPv6 address of the neighbor> capability graceful-restart	NS(config-router-af)#neighbor 2001:db8::10 capability graceful-restart	Enables graceful restart with the specified neighbor router device.
redistribute kernel	NS(config-router-af)#redistribute kernel	Redistributes kernel routes.
exit-address-family	NS(config-router-af)#exit-address-family	Exits address family configuration mode.

Configuring IPv6 RIP

September 21, 2020

IPv6 Routing Information Protocol (RIP) or RIPng is a Distance Vector protocol. This protocol is an extension of RIP to support IPv6. After enabling IPv6 RIP, you need to configure advertisement of IPv6 RIP routes. For troubleshooting, you can limit IPv6 RIP propagation. You can display IPv6 RIP settings to verify the configuration.

Prerequisites for IPv6 RIP

Before you begin configuring IPv6 RIP, do the following:

- Make sure that you understand the IPv6 RIP protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Advertising IPv6 RIP Routes

IPv6 RIP enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertisement enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 RIP to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting IPv6 RIP Propagations

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given interface.

To limit IPv6 RIP propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 rip	Start IPv6 RIP routing process and enter configuration mode for the routing process.
passive-interface < vlan_name>	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 rip
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the IPv6 RIP Configuration

You can use VTYSH to display the IPv6 RIP routing table and IPv6 RIP information for a specified VLAN.

To view the IPv6 RIP settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.

Commands	Specifies
sh ipv6 rip	Display updated IPv6 RIP routing table.
sh ipv6 rip interface <vlan_name>	Display IPv6 RIP information for the specified VLAN.

Example:

```
NS# VTYSH
NS# sh ipv6 rip
NS# sh ipv6 rip interface VLAN0
<!--NeedCopy-->
```

Configuring IPv6 OSPF

September 21, 2020

IPv6 OSPF or OSPF version 3 (OSPF v3) is a link state protocol that is used to exchange IPv6 routing information. After enabling IPv6 OSPF, you need to configure advertisement of IPv6 OSPF routes. For troubleshooting, you can limit IPv6 OSPF propagation. You can display IPv6 OSPF settings to verify the configuration.

Prerequisites for IPv6 OSPF

Before you begin configuring IPv6 OSPF, do the following:

- Make sure that you understand the IPv6 OSPF protocol.
- Install the IPv6PT license on the NetScaler appliance.
- Enable the IPv6 feature.

Advertising IPv6 Routes

IPv6 OSPF enables an upstream router to load balance traffic between two identical vservers hosted on two standalone NetScaler devices. Route advertising enables an upstream router to track network entities located behind the NetScaler.

To configure IPv6 OSPF to advertise IPv6 routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
redistribute static	Redistribute static routes.
redistribute kernel	Redistribute kernel routes.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# redistribute static
NS(config-router)# redistribute kernel
<!--NeedCopy-->
```

Limiting IPv6 OSPF Propagations

If you need to troubleshoot your configuration, you use VTYSH to configure listen-only mode on any given VLAN.

To limit IPv6 OSPF propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
router ipv6 OSPF	Start IPv6 OSPF routing process and enter configuration mode for the routing process.
passive-interface < vlan_name >	Suppress routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
```

```
NS# configure terminal
NS(config)# router ipv6 OSPF
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the IPv6 OSPF Configuration

You use VTYSH to display IPv6 OSPF current neighbors and IPv6 OSPF routes.

To view the IPv6 OSPF settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Specifies
VTYSH	Display VTYSH command prompt.
sh ipv6 OSPF neighbor	Display current neighbors.
sh ipv6 OSPF route	Display IPv6 OSPF routes.

Example:

```
>VTYSH
NS# sh ipv6 OSPF neighbor
NS# sh ipv6 OSPF route
<!--NeedCopy-->
```

OSPFv3 Authentication

To ensure the integrity, data origin authentication, and data confidentiality of OSPFv3 packets, OSPFv3 authentication must be configured on OSPFv3 peers.

The NetScaler appliance supports OSPFv3 authentication and is partially compliant with RFC 4552. OSPFv3 authentication is based on the two IPSec protocols: Authentication Header (AH) and Encapsulating Security Payload (ESP). The NetScaler appliance supports only the AH protocol for OSPFv3 authentication.

OSPFv3 authentication uses manually defined IPSec Security Associations (SAs) between the OSPFv3 peers and does not rely on IKE protocol for forming dynamic SAs. Manual SAs define the security parameter Index (SPI) values, algorithms, and keys to be used between the peers. Manual SAs require no negotiation between the peers; therefore, the same SA must be defined on both the peers.

You can configure OSPFv3 authentication on a VLAN or for an OSPFv3 area. When you configure for a

VLAN, the settings are applied to all the interfaces that are members of the VLAN. When you configure OSPFv3 authentication for an OSPF area, the settings are applied to all the VLANs in that area. The settings are in turn applied to all the interfaces that are members of these VLANs. These settings do not apply to member VLANs on which you have configured OSPFv3 authentication directly.

Consider the following points and limitations before configuring OSPFv3 authentication on a NetScaler appliance:

- Make sure that you understand the different components of OSPFv3 authentication, described in RFC 4552.
- Only Authentication Header protocol is supported for OSPFv3 authentication. Encapsulating Security Payload (ESP) is not supported.
- You must define an SA with the same setting on the peer interface.
- Rekeying of manual keys is not supported.

To configure OSPFv3 authentication on a VLAN by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown: OSPFv3 authentication VLAN commands.

Example:

```
> VTYSH NS# configure terminal
NS(config)# interface vlan2
NS(config-if)# ipv6 ospf authentication ipsec spi 256 md5 123456789
    ABCDEF0123456789ABCDEF0
<!--NeedCopy-->
```

To configure OSPFv3 authentication on an OSPF area by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown: OSPFv3 authentication OSPF area commands.

Example:

```
> VTYSH NS# configure terminal
ns(config)#router ipv6 ospf 30
ns(config-router)# area 1 authentication ipsec spi 256
    md5123456789ABCDEF0123456789ABCDEF0
<!--NeedCopy-->
```

Configuring Graceful Restart for IPv6 OSPF

In a non-INC high availability (HA) setup in which a routing protocol is configured, after a failover, routing protocols are converged and routes between the new primary node and the adjacent neighbor routers are learned. Route learning take some time to complete. During this time, forwarding of packets is delayed, network performance might get disrupted, and packets might get dropped.

Graceful restart enables an HA setup during a failover to direct its adjacent routers to not remove the old primary node's learned routes from their routing databases. Using the old primary node's routing information, the new primary node and the adjacent routers immediately start forwarding packets, without disrupting network performance.

To configure graceful restart for IPv6 OSPF by using the VTYSH command line, at the command prompt, type the following commands, in the order shown:

Command	Example	Command Description
VTYSH	> VTYSH	Enters VTYSH command prompt.
configure terminal	NS# configure terminal	Enters global configuration mode.
router-id id>	NS(config)#router-id 1.1.1.1	Sets a router identifier for the NetScaler appliance. This identifier is set for all the dynamic routing protocols. The same ID must be specified in the other node in a high availability set up for graceful restart to work properly in the HA set up.
IPv6ospf restart grace-period <1-1800>	NS(config)# IPv6ospf restart grace-period 170	Specifies the grace period, in seconds, for which the routes are to be preserved in the helper devices. Default value: 120 seconds.

Command	Example	Command Description
IPv6 ospf restart helper max-grace-period <1-1800>	NS(config)# IPv6 ospf restart helper max-grace-period 180	This is an optional command to limit the maximum grace period for which the NetScaler appliance will be in the helper mode. If the NetScaler appliance receives an opaque LSA with grace-period greater than the set helper max-grace-period, the LSA is discarded and the NetScaler is not placed in helper mode.
interface <VLANID>	NS(config)#interface vlan3	Enters VLAN configuration mode.
ipv6 router ospf area tag	NS(config-if)#ipv6 router ospf area 0 tag 1	Starts IPv6 OSPF routing process on a VLAN.
exit	NS(config-if)#exit	Exits VLAN configuration mode.
router ipv6 ospf	NS(config)# router ipv6 ospf 1	Starts IPv6 OSPF routing process and enters configuration mode for the routing process.
capability restart graceful	NS(config-router)#capability restart graceful	Enables graceful restart on the IPv6 OSPF routing process.
redistribute kernel	NS(config-router)# redistribute kernel	Redistributes kernel routes.

Configuring ISIS

September 21, 2020

The NetScaler appliance supports the Intermediate System-to-Intermediate System (IS-IS or ISIS) dynamic routing protocol. This protocol supports IPv4 as well as IPv6 route exchanges. IS-IS is a link state protocol and is therefore less prone to routing loops. With the advantages of faster convergence and the ability to support larger networks, ISIS can be very useful in Internet Service Provider (ISP) networks.

Prerequisites for configuring ISIS

Before you begin configuring ISIS, do the following:

- Make sure that you understand the ISIS protocol.
- For IPV6 routes, enable:
    - IPv6 protocol translation feature.
    - IPv6 Dynamic Routing option on the VLANs on which you want to run ISIS protocol.

Enabling ISIS

Use either of the following procedures to enable the ISIS routing feature on the NetScaler appliance.

To enable ISIS routing by using the NetScaler command line:

At the command prompt, type:

enable ns feature ISIS

To enable ISIS routing by using the NetScaler GUI:

1. Navigate to System > Settings, in Modes and Features group, click Change advanced features.
2. Select or clear the ISIS Routing option.

Creating an ISIS Routing Process and Starting It on a VLAN

To create an ISIS routing process, you must use the VTYSH command line.

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Creates an ISIS routing process and configuration mode for the routing process.

Command	Description		
net XX….XXXX.YYYY.YYYY.YYYY.00	Specifies a NET value for the routing process, where: · **XX. .. .XXXX** is the Area Address (can be 1-13 bytes), · **YYYY.YYYY.YYYY** is the System ID (6 bytes), · **00** is the N-selector (1 byte).		
is-type (level-1	level-1-2	level-2-only)	Sets the ISIS routing process to the specified level of routing. Default: level-1-2.
ns IPv6-routing	Starts the IPv6 dynamic routing daemon.		
interface	Enters the VLAN configuration mode.		
ip router ISIS	Enables the ISIS routing process on the VLAN for IPv4 route exchanges.		
ipv6 router ISIS	Enables the ISIS routing process on the VLAN for IPv6 route exchanges.		

Example:

```
> VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# net 15.aabb.ccdd.0097.00
NS(config-router)# is-type level-1
NS(config-router)# exit
NS(config)# ns IPv6-routing
NS(config)# interface vlan0
NS(config-if)#  ip router isis 11
NS(config-if)# ipv6 router isis 11
<!--NeedCopy-->
```

Advertising Routes

Route advertisement enables an upstream router to track network entities located behind the NetScaler appliance.

To configure ISIS to advertise routes by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router ISIS [tag]	Starts the ISIS routing instance and enter configuration mode for the routing process.
redistribute connected (level-1 or level-1-2 or level-2)	Redistributes connected routes, where: **level-1**: Redistribute connected routes into Level-1, **level-1-2**: Redistribute connected routes into Level-1 and Level-2, **level-2**: Redistribute connected routes into Level-2.
redistribute kernel (level-1 or level-1-2 or level-2)	Redistributes kernel routes, where: **level-1**: Redistribute kernel routes into Level-1, **level-1-2**: Redistribute kernel routes into Level-1 and Level-2, **level-2**: Redistribute kernel routes into Level-2.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# redistribute connected level-1
NS(config-router)# redistribute kernel level-1
<!--NeedCopy-->
```

Limiting ISIS Propagations

If you need to troubleshoot your configuration, you can configure the listen-only mode on any given VLAN.

To limit ISIS propagation by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Command	Description
VTYSH	Displays the VTYSH command prompt.
configure terminal	Enters the global configuration mode.
router isis [tag]	Enters the configuration mode for the routing process.
passive-interface <vlan_name>	Suppresses routing updates on interfaces bound to the specified VLAN.

Example:

```
>VTYSH
NS# configure terminal
NS(config)# router isis 11
NS(config-router)# passive-interface VLAN0
<!--NeedCopy-->
```

Verifying the ISIS Configuration

You can use VTYSH to display the ISIS routing table and ISIS information for a specified VLAN.

To view the ISIS settings by using the VTYSH command line:

At the command prompt, type the following commands, in the order shown:

Commands	Description
VTYSH	Displays the VTYSH command prompt.
show ip isis route	Displays updated IPv4 ISIS routing table.
show ipv6 isis route	Displays updated IPv6 ISIS routing table.
sh isis interface <vlan_name>	Displays IPv6 ISIS information for the specified VLAN.

Example:

```
NS# VTYSH
NS#  show ip isis route
NS#  show ipv6 isis route
NS# sh isis interface VLAN0
<!--NeedCopy-->
```

Install Routes to the NetScaler Routing Table

September 21, 2020

The NetScaler appliance can use routes learned by various routing protocols after you install the routes in the appliance's routing table.

To install various routes to the internal routing table by using the VTYSH command line:

At the NetScaler command line, type the following commands as appropriate for the routes that you want to install:

Commands	Specifies
VTYSH	Display VTYSH command prompt.
configure terminal	Enter global configuration mode.
ns route-install Default	Install IPv4 default routes to the internal routing table.
ns route-install RIP	Install IPv4 RIP specific routes to the internal routing table.
ns route-install BGP	Install IPv4 BGP specific routes to the internal routing table.
ns route-install OSPF	Install IPv4 OSPF specific routes to the internal routing table.
ns route-install IPv6 Default	Install IPv6 default routes to the internal routing table.
ns route-install IPv6 RIP	Install IPv6 RIP specific routes to the internal routing table.
ns route-install IPv6 BGP	Install IPv6 BGP specific routes to the internal routing table.

Commands	Specifies
ns route-install IPv6 OSPF	Install IPv6 OSPF specific routes to the internal routing table.

Example:

```
>VTYSH
NS# configure terminal
NS# ns route-install Default
NS(config)# ns route-install RIP
NS(config)# ns route-install BGP
NS(config)# ns route-install OSPF
NS# ns route-install IPv6 Default
NS(config)# ns route-install IPv6 RIP
NS(config)# ns route-install IPv6 BGP
NS(config)# ns route-install IPv6 OSPF
<!--NeedCopy-->
```

Advertisement of SNIP and VIP Routes to Selective Areas

September 21, 2020

To advertise some SNIP addresses to selective areas, enabling DRADV mode or redistribute connect ZebOS operations cannot be used. This is because these operations send all the connected routes to ZebOS. Also, adding dummy static routes in ZebOS for the required subnets, or adding ACLs in ZebOS to filter unwanted connected routes, is a cumbersome and tedious task.

The Network Route and the Tag options address this issue. You can enable the Network Route option for only one SNIP address per subnet. The connected route for that SNIP address is sent as a kernel route to ZebOS.

For VIP and SNIP addresses, Tag, can be assigned an integer from 1 to 4294967295. This parameter can be set only when Host Route or Network Route is enabled for VIP or SNIP addresses. The tag value associated with VIP and SNIP addresses are also sent along with their routes to ZebOS. Tags with different values can be set for VIP and SNIP routes. These tag values can then be matched in route maps in ZebOS and advertised to selective areas.

Advertise SNIP Routes to Selective Areas

To configure the network route and tag parameters of a SNIP address by using the NetScaler command line:

At the command prompt, type:

- If adding a new SNIP address:
  - **add ns ip** <IPAddress>@ <netmask> -**type SNIP** -**networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- If reconfiguring an existing SNIP address:
  - **set ns ip** <IPAddress>@ <netmask> -**type SNIP** - **networkroute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

To configure the network route and tag parameters of a SNIP address by using NetScaler GUI:

1. Navigate to **System**> **Network** > **IPs** > **IPV4s**.
2. Set the **Network Route** and **Tag** parameters while adding a Subnet IP (SNIP) address or modifying an existing Subnet IP address.

Advertise VIP Routes to Selective Areas

To configure the host route and tag parameters of a VIP address by using the NetScaler command line:

At the command prompt, type one of the following sets of commands.

- If adding a new VIP address:
  - **add ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>
- If reconfiguring an existing VIP address:
  - **set ns ip** <IPAddress>@ <netmask> -**type VIP** -**hostRoute** (**ENABLED** | **DISABLED**) -**tag** <positive_integer>
  - **show ns ip** <IPAddress>

To configure the network route and tag parameters of a VIP address by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** > **IPV4s**.
2. Set the **Host Route** and **Tag** parameters while adding a VIP address or modifying an existing VIP address.

Configuring Static Routes

September 21, 2020

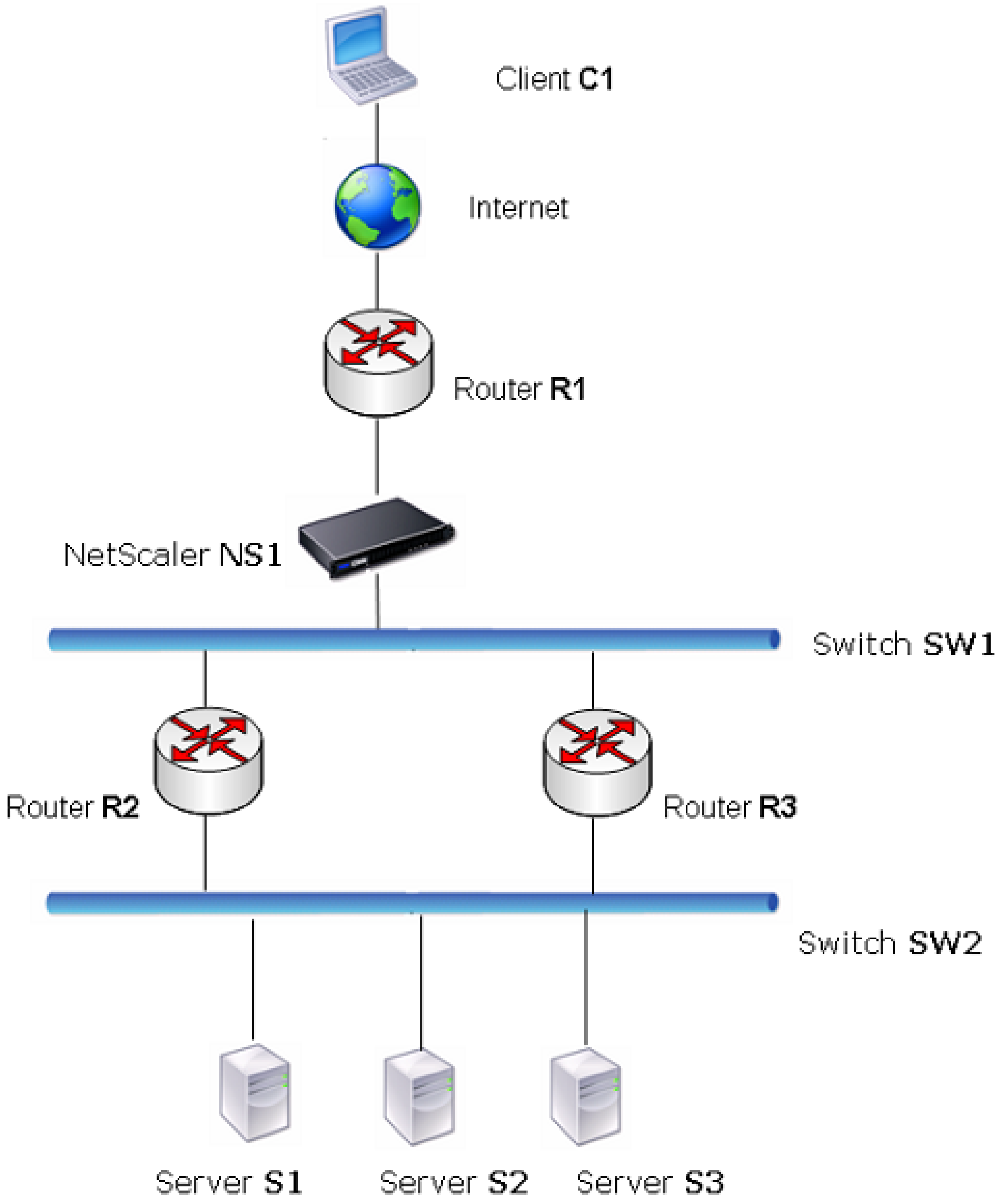
Static routes are manually created to improve the performance of your network. You can monitor static routes to avoid service disruptions. Also, you can assign weights to ECMP routes, and you can create null routes to prevent routing loops.

Monitored Static Routes. If a manually created (static) route goes down, a backup route is not automatically activated. You must manually delete the inactive primary static route. However, if you configure the static route as a monitored route, the NetScaler appliance can automatically activate a backup route.

Static route monitoring can also be based on the accessibility of the subnet. A subnet is usually connected to a single interface, but it can be logically accessed through other interfaces. Subnets bound to a VLAN are accessible only if the VLAN is up. VLANs are logical interfaces through which packets are transmitted and received by the NetScaler. A static route is marked as DOWN if the next hop resides on a subnet that is unreachable.

Note: In a high availability (HA) setup, the default value for monitored state routes (MSRs) on the secondary node is UP. The value is set to avoid a state transition gap upon failover, which could result in dropping packets on those routes.

Consider the following simple topology, in which a NetScaler is load balancing traffic to a site across multiple servers.


Client C1
Internet
Router R1
NetScaler NS1
Switch SW1
Router R2
Router R3
Switch SW2
Server S1
Server S2
Server S3

Router R1 moves traffic between the client and the NetScaler appliance. The appliance can reach servers S1 and S2 through routers R2 or R3. It has two static routes through which to reach the servers' subnet, one with R2 as the gateway and another with R3 as the gateway. Both these routes have monitoring enabled. The administrative distance of the static route with gateway R2 is lower than that of the static route with gateway R3. Therefore, R2 is preferred over R3 to forward traffic to the servers. Also, the default route on the NetScaler points to R1 so that all Internet traffic exits properly.

If R2 fails while monitoring is enabled on the static route, which uses R2 as the gateway, the NetScaler marks it as DOWN. The NetScaler now uses the static route with R3 as the gateway and forwards the traffic to the servers through R3.

The NetScaler supports monitoring of IPv4 and IPv6 static routes. You can configure the NetScaler to monitor an IPv4 static route either by creating a new ARP or PING monitor or by using existing ARP or PING monitors. You can configure the NetScaler to monitor an IPv6 static route either by creating a new Neighbor discovery for IPv6 (ND6) or PING monitor or by using the existing ND6 or PING monitors.

Weighted Static Routes. When the NetScaler appliance makes routing decisions involving routes with equal distance and cost, that is, Equal Cost Multi-Path (ECMP) routes, it balances the load between them by using a hashing mechanism based on the source and destination IP addresses. For an ECMP route, however, you can configure a weight value. The NetScaler then uses both the weight and the hashed value for balancing the load.

Null Routes. If the route chosen in a routing decision is inactive, the NetScaler appliance chooses a backup route. If all the backup routes become inaccessible, the appliance might reroute the packet to the sender, which could result in a routing loop leading to network congestion. To prevent this situation, you can create a null route, which adds a null interface as a gateway. The null route is never the preferred route, because it has a higher administrative distance than the other static routes. But it is selected if the other static routes become inaccessible. In that case, the appliance drops the packet and prevents a routing loop.

Configuring IPv4 Static Routes

You can add a simple static route or a null route by setting a few parameters, or you can set additional parameters to configure a monitored or monitored and weighted static route. You can change the parameters of a static route. For example, you might want to assign a weight to an unweighted route, or you might want to disable monitoring on a monitored route.

NetScaler command line procedures

To create a static route by using the NetScaler command line:

At the command prompt, type:

- add route <network> <netmask> <gateway>[-cost <positive_integer>] [-advertise (DISABLED | ENABLED)]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example:

```
> add route 10.102.29.0 255.255.255.0 10.102.29.2 -cost 2 -advertise
   ENABLED
 Done
<!--NeedCopy-->
```

To create a monitored static route by using the NetScaler command line:

At the command prompt, type the following commands to create a monitored static route and verify the configuration:

- add route <network> <netmask> <gateway> [-distance <positive_integer>] [-weight <positive_integer>][-msr (ENABLED | DISABLED) [-monitor <string>]]
- show route [<network> <netmask> [<gateway>]] [<routeType>] [-detail]

Example:

```
> add route 10.102.29.0 255.255.255.0 10.102.29.3 -distance 5 -weight 6
    -msr ENBLED -monitor PING
 Done
<!--NeedCopy-->
```

To create a null route by using the NetScaler command line:

At the command prompt type:

- add route <network> <netmask> null
- show route <network> <netmask>

Example:

```
> add route 10.102.29.0 255.255.255.0 null
 Done
<!--NeedCopy-->
```

To remove a static route by using the NetScaler command line:

At the command prompt, type:

rm route <network> <netmask> <gateway>

Example:

```
> rm route 10.102.29.0 255.255.255.0 10.102.29.3
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To configure a static route by using the NetScaler GUI:

Navigate to System > Network > Routes and, on the Basic tab, add a new static route, or edit an existing static route.

To remove a route by using the NetScaler GUI:

Navigate to System > Network > Routes and, on the Basic tab, delete the static route.

Configuring IPv6 Static Routes

You can configure a maximum of six default IPv6 static routes. IPv6 routes are selected on the basis of whether the MAC address of the destination device is reachable. This can be determined by using the IPv6 Neighbor Discovery feature. Routes are load balanced and only source/destination-based hash mechanisms are used. Therefore, route selection mechanisms such as round robin are not supported. The next hop address in the default route need not belong to the NSIP subnet.

NetScaler command line procedures

To create an IPv6 route by using the NetScaler command line:

At the command prompt, type the following commands to create an IPv6 route and verify the configuration:

- add route6 <network> <gateway> [-vlan <positive_integer>]
- show route6 [<network> [<gateway>]

Example:

```
> add route6 ::/0 FE80::67 -vlan 5
 Done
<!--NeedCopy-->
```

To create a monitored IPv6 static route by using the NetScaler command line:

At the command prompt, type the following commands to create a monitored IPv6 static route and verify the configuration:

- add route6 <network> <gateway> [-msr (ENABLED | DISABLED) [-monitor <string>]
- show route6 [<network> [<gateway>]

Example:

```
> add route6 ::/0 2004::1 -msr ENABLED -monitor PING
 Done
<!--NeedCopy-->
```

To remove an IPv6 route by using the NetScaler command line:

At the command prompt, type:

rm route6 <network> <gateway>

Example:

```
> rm route6 ::/0 FE80::67
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To configure an IPv6 route by using the NetScaler GUI:

Navigate to System > Network > Routes and, on the IPV6 tab, add a new IPv6 route, or edit an existing IPv6 route.

To remove an IPv6 route by using the NetScaler GUI:

Navigate to System > Network > Routes and, on the IPV6 tab, delete the IPv6 route.

Route Health Injection Based on Virtual Server Settings

September 21, 2020

The following option and parameter are introduced for controlling the Route Health Injection (RHI) functionality of the NetScaler appliance for advertising the route of a VIP address.

- **VSVR_CNTRLD**. It is an option for the (Vserver RHI Level) parameter of a VIP address. When this option is set to the Vserver RHI Level parameter, the RHI behavior for advertising the route of the VIP address depends on the RHI STATE parameter setting on all the associated virtual servers of the VIP address along with their states.
- **RHI STATE**. It is a parameter of virtual server. You can set the RHI STATE parameter to either PASSIVE or ACTIVE. By default, the RHI STATE parameter is set to PASSIVE.

For a VIP address, when RHI (Vserver RHI Level) parameter is set to VSVR_CNTRLD, the following are different RHI behaviours for the VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address:

- If you set RHI STATE to PASSIVE on all virtual servers, the NetScaler appliance always advertises the route for the VIP address.
- If you set RHI STATE to ACTIVE on all virtual servers, the NetScaler appliance advertises the route for the VIP address if at least one of the associated virtual servers is in UP state.

- If you set RHI STATE to ACTIVE on some and PASSIVE on others, the NetScaler appliance advertises the route for the VIP address if at least one of the associated virtual servers, whose RHI STATE set to ACTIVE, is in UP state.

Following table displays the sample RHI behaviour for a VIP address on the basis of RHI STATE settings on the virtual servers associated with the VIP address. The NetScaler appliance has two virtual servers V1 and V2 associated with the VIP address:

Associated virtual servers for a VIP	State 1	State 2	State 3	State 4
RHI State set to PASSIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	Yes
RHI State set to ACTIVE on all virtual servers				
V1	UP	UP	DOWN	DOWN
V2	UP	DOWN	UP	DOWN
Advertise the route for this VIP address?	Yes	Yes	Yes	No
RHI State set to ACTIVE on one virtual server and PASSIVE on the other				
V1 (RHI State = ACTIVE)	UP	UP	DOWN	DOWN
V2 (RHI State = PASSIVE)	UP	DOWN	UP	DOWN

Associated virtual servers for a VIP	State 1	State 2	State 3	State 4
Advertise the route for this VIP address?	Yes	Yes	No	No

To configure RHI for a VIP address, to be based on the RHI (RHI State) parameter setting of the associated virtual servers, perform the following steps:

- Set the RHI (Vserver RHI Level) parameter to VSVR_CNTRLD for the VIP address.
- Set the RHI State parameter for each virtual server associated with the VIP address.

To set the vServer RHI Level for a VIP address by using the NetScaler command line:

At the command prompt, type:

- **set ns ip** <IPAddress> [-**vserverRHILevel** <vserverRHILevel>]

To set the RHI State parameter of a virtual server by using the NetScaler command line:

At the command prompt, type:

- **set lb vserver** <name> [-**RHIstate** (**PASSIVE** | **ACTIVE**)]

To set the vServer RHI Level for a VIP address by using NetScaler GUI

1. Navigate to **System** > **Network** > **IPs**.
2. Select a VIP address, and then click **Edit**.
3. Set the **Vserver RHI Level** parameter to **VSVR_CNTRLD**, and then click **OK**.

To set the RHI State parameter of a virtual server by using NetScaler GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Select a load balancing virtual server, and then click **Edit**.
3. Set the **RHI State** parameter, and then click **OK**.

Configuring Policy-Based Routes

September 21, 2020

Policy-based routing bases routing decisions on criteria that you specify. A policy-based route (PBR) specifies criteria for selecting packets and, typically, a next hop to which to send the selected packets. For example, you can configure the NetScaler appliance to route outgoing packets from a specific IP address or range to a particular next hop router. Each packet is matched against each configured PBR, in the order determined by the specified priorities, until a match is found. If no match is found, or if the matching PBR specifies a DENY action, the NetScaler applies the routing table for normal destination-based routing.

A PBR bases routing decisions for the data packets on parameters such as source IP address, source port, destination IP address, destination port, protocol, and source MAC address. A PBR defines the conditions that a packet must satisfy for the NetScaler to route the packet. These actions are known as "processing modes." The processing modes are:

- **ALLOW.** The NetScaler sends the packet to the designated next-hop router.
- **DENY.** The NetScaler applies the routing table for normal destination-based routing.

You can create PBRs for outgoing IPv4 and IPv6 traffic.

Many users begin by creating PBRs and then modifying them. To activate a new PBR, you must apply it. To deactivate a PBR, you can either remove or disable it. You can change the priority number of a PBR to give it a higher or lower precedence.

Policy-Based Routes (PBR) for IPv4 Traffic

September 21, 2020

Configuring PBRs involves the following tasks:

- Create a PBR.
- Apply PBRs.
- (Optional) Disable or enable a PBR.
- (Optional) Renumber the priority of the PBR.

Creating or Modifying a PBR

You cannot create two PBRs with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR. The priority (an integer value) defines the order in which the NetScaler appliance evaluates PBRs. When you create a PBR without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR, the NetScaler performs an action. If the packet does not match the condition defined by the PBR, the NetScaler compares the packet against the PBR with the next highest priority.

Instead of sending the selected packets to a next hop router, you can configure the PBR to send them to a link load balancing virtual server to which you have bound multiple next hops. This configuration can provide a backup if a next hop link fails.

Consider the following example. Two PBRs, p1 and p2, are configured on the NetScaler and automatically assigned priorities 20 and 30. You need to add a third PBR, p3, to be evaluated immediately after the first PBR, p1. The new PBR, p3, must have a priority between 20 and 30. In this case, you can specify the priority as 25.

NetScaler command line procedures

To create a PBR by using the NetScaler command line:

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr

Example:

```
> add ns pbr pbr1  allow -srcip 10.102.37.252 -destip 10.10.10.2 -
   nexthop 10.102.29.77
 Done
<!--NeedCopy-->
```

To modify the priority of a PBR by using the NetScaler command line:

At the command prompt, type the following commands to modify the priority and verify the configuration:

- set ns pbr <name> [-action (ALLOW | DENY)] [-srcIP [<operator>] <srcIPVal>] [-srcPort [<operator>] <srcPortVal>] [-destIP [<operator>] <destIPVal>] [-destPort [<operator>] <destPortVal>] [-nextHop <nextHopVal>] [-srcMac <mac_addr>] [-protocol <protocol> | -protocolNumber <positive_integer>] [-vlan <positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-state (ENABLED | DISABLED)]
- show ns pbr [<name>]

Example:

```
> set ns pbr pbr1 -priority 23
 Done
```

```
<!--NeedCopy-->
```

To remove one or all PBRs by using the NetScaler command line:

At the command prompt, type one of the following commands:

- rm ns pbr <name>
- clear ns pbrs

Example:

```
> rm ns pbr pbr1
 Done

> clear ns PBRs
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To create a PBR by using the NetScaler GUI:

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR, or edit an existing PBR.

To remove one or all PBRs by using the NetScaler GUI:

Navigate to System > Network > PBRs, on the PBRs tab, delete the PBR.

Applying a PBR

You must apply a PBR to activate it. The following procedure reapplies all PBRs that you have not disabled. The PBRs constitute a memory tree (lookup table). For example, if you create 10 PBRs (p1 - p10), and then you create another PBR (p11) and apply it, all of the PBRs (p1 - p11) are freshly applied and a new lookup table is created. If a session has a DENY PBR related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR. For example, you must follow this procedure after disabling a PBR.

Note: PBRs created on the NetScaler appliance do not work until they are applied.

To apply a PBR by using the NetScaler command line:

At the command prompt, type:

apply ns PBRs

To apply a PBR by using the NetScaler GUI:

1. Navigate to System > Network > PBRs.

2. On the PBRs tab, select the PBR, in the Action list, select Apply.

Enabling or Disabling PBRs

By default, the PBRs are enabled. This means that when PBRs are applied, the NetScaler appliance automatically compares incoming packets against the configured PBRs. If a PBR is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBRs are applied. After the PBRs are applied, the NetScaler does not compare incoming packets against disabled PBRs.

To enable or disable a PBR by using the NetScaler command line:

At the command prompt, type one of the following commands:

- enable ns pbr <name>
- disable ns pbr <name>

Example:

```
> enable ns PBR pbr1
 Done
> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                          Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                Protocol:
        Vlan:                                  Interface:
        Active Status: ENABLED                 Applied Status: APPLIED
        Priority: 10
        NextHop: 10.102.29.77

 Done

> disable ns PBR pbr1
Warning: PBR modified, use 'apply pbrs' to commit this operation

> apply pbrs
 Done

> show ns PBR pbr1
1)      Name: pbr1
        Action: ALLOW                          Hits: 0
        srcIP = 10.102.37.252
        destIP = 10.10.10.2
        srcMac:                                Protocol:
```

```
        Vlan:                                  Interface:
        Active Status: DISABLED                Applied Status: NOTAPPLIED
        Priority: 10
        NextHop: 10.102.29.77
Done
<!--NeedCopy-->
```

To enable or disable a PBR by using the NetScaler GUI:

1. Navigate to System > Network > PBRs.
2. On the PBRs tab, select the PBR, in the Action list, select Enable or Disable.

Renumbering PBRs

You can automatically renumber the PBRs to set their priorities to multiples of 10.

To renumber PBRs by using the NetScaler command line:

At the command prompt, type:

- renumber ns pbrs

To renumber PBRs by using the NetScaler GUI:

Navigate to System > Network > PBRs, on the PBRs tab, in the Action list, select Renumber Priority (s).

Use Case - PBR with Multiple Hops

Consider a scenario in which two PBRs, PBR1 and PBR2, are configured on NetScaler appliance NS1. PBR1 routes all the outgoing packets, with source IP address as 10.102.29.30, to next hop router R1. PBR2 routes all the outgoing packets, with source IP address as 10.102.29.90, to next hop router R2. R3 is another next hop router connected to NS1.

If router R1 fails, all the outgoing packets that matched against PBR1 are dropped. To avoid this situation, you can specify a link load balancing (LLB) virtual server in the next hop field while creating or modifying a PBR. Multiple next hops are bound to the LLB virtual server as services (for example R1, R2, and R3). Now, if R1 fails, all the packets that matched against PBR1 are routed to R2 or R3 as determined by the LB method configured on the LLB virtual server.

The NetScaler appliance throws an error if you attempt to create a PBR with an LLB virtual server as the next hop in the following cases:

- Adding another PBR with the same LLB virtual server.
- Specifying a nonexistent LLB virtual server.
- Specifying an LLB virtual server for which the bound services are not next hops.

- Specifying an LLB virtual server for which the LB method is not set to one of the following:
  - LEASTPACKETS
  - LEASTBANDWIDTH
  - DESTIPHASH
  - SOURCEIPHASH
  - WEIGHTDRR
  - SRCIPDESTIP_HASH
  - LTRM
  - CUSTOM LOAD
- Specifying an LLB virtual server for which the LB persistence type is not set to one of the following:
  - DESTIP
  - SOURCEIP
  - SRCDSTIP

The following table lists the names and values of the entities configured on the NetScaler appliance:

Entity Type	Name	IP Address
Link load balancing virtual server	LLB1	NA
Services (next hops)	Router1	1.1.1.254
	Router2	2.2.2.254
	Router3	3.3.3.254
PBRs	PBR1	NA
	PBR2	NA

Table 1. Sample Values for Creating Entities

To implement the configuration described above, you need to:

1. Create services Router1, Router2, and Router3 that represent next hop routers R1, R2, and R3.
2. Create link load balancing virtual server LLB1 and bind services Router1, Router2, and Router3 to it.
3. Create PBRs PBR1 and PBR2, with next hop fields set as LLB1 and 2.2.2.254 (IP address of the router R2), respectively.

To create a service by using the NetScaler command line:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port>

- show service <name>

Example:

```
> add service Router1 1.1.1.254 ANY *
 Done
> add service Router2 2.2.2.254 ANY *
 Done
> add service Router3 3.3.3.254 ANY *
 Done
<!--NeedCopy-->
```

To create a service by using the NetScaler GUI:

Navigate to Traffic Management > Load Balancing > Services, and create a service.

To create a link load balancing virtual server and bind a service by using the NetScaler command line:

At the command prompt, type:

- add lb vserver <name> <serviceType>
- bind lb vserver < name> <serviceName>
- show lb vserver < name>

Example:

```
> add lb vserver LLB1 ANY
 Done
> bind lb vserver LLB1 Router1 Router2 Router3
 Done
<!--NeedCopy-->
```

To create a link load balancing virtual server and bind a service by using the NetScaler GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and create a virtual server for link load balancing. Specify **ANY** in the **Protocol** field.
   Note: Make sure that
   Directly Addressable is unchecked.
2. Under the **Services** tab, in the **Active** column, select the check box for the service that you want to bind to the virtual server.

To create a PBR by using the NetScaler command line:

At the command prompt, type:

- add ns pbr <name> <action> [-srcIP [<operator>] <srcIPVal>] [-nextHop <nextHopVal>]
- show ns pbr

Example:

```
> add pbr PBR1 ALLOW -srcIP 10.102.29.30 -nextHop LLB1
 Done
> add pbr PBR2 ALLOW -srcIP 10.102.29.90 -nextHop 2.2.2.254
 Done
<!--NeedCopy-->
```

To create a PBR by using the NetScaler GUI:

Navigate to System > Network > PBRs, on the PBRs tab, add a new PBR.

Policy-Based Routes (PBR6) for IPv6 Traffic

September 21, 2020

Configuring PBR6s involves the following tasks:

- Create a PBR6.
- Apply PBR6s.
- (Optional) Disable or enable a PBR6.
- (Optional) Renumber the priority of the PBR6.

Creating or Modifying a PBR6

You cannot create two PBR6s with the same parameters. If you attempt to create a duplicate, an error message appears.

You can configure the priority of a PBR6. The priority (an integer value) defines the order in which the NetScaler appliance evaluates PBR6s. When you create a PBR6 without specifying a priority, the NetScaler automatically assigns a priority that is a multiple of 10.

If a packet matches the condition defined by the PBR6, the NetScaler performs an action. If the packet does not match the condition defined by the PBR6, the NetScaler compares the packet against the PBR6 with the next highest priority.

NetScaler command line procedures

To create a PBR6 by using the NetScaler command line:

At the command prompt, type:

- **add ns pbr6** <name> <action> [-srcIPv6 [<operator>] <srcIPv6Val>] [-srcPort [<operator>] <srcPortVal>] [-destIPv6 [<operator>] <destIPv6Val>] [-destPort [<operator>] <destPortVal>] [-srcMac <mac_addr>] [-protocol <protocol> |-protocolNumber <positive_integer>] [-vlan

<positive_integer>] [-interface <interface_name>] [-priority <positive_integer>] [-state (ENABLED | DISABLED)] [-msr (ENABLED | DISABLED) [-monitor <string>]] [-nextHop <nextHopVal>] [-nextHopVlan <positive_integer>]
- show ns pbr

To modify or remove a PBR6 by using the NetScaler command line:

To modify a PBR6, type the **set pbr6** <name> command and the parameters to be changed, with their new values.

To remove one or all PBR6s by using the NetScaler command line:

At the command prompt, type one of the following commands:

- **rm ns pbr6** <name>
- **clear ns pbr6**

NetScaler GUI procedures

To create or modify a PBR6 by using the NetScaler GUI:

Navigate to System > Network > PBRs and, on the PBR6s tab, add a new PBR6, or edit an existing PBR6.

To remove one or all PBR6s by using the NetScaler GUI:

Navigate to System > Network > PBRs and, on the PBR6s tab, delete the PBR6.

Applying PBR6s

You must apply a PBR6 to activate it. The following procedure reapplies all PBR6s that you have not disabled. The PBR6s constitute a memory tree (lookup table). For example, if you create 10 PBR6s (p6_1 - p6_10), and then you create another PBR6 (p6_11) and apply it, all of the PBR6s (p6_1 - p6_11) are freshly applied and a new lookup table is created. If a session has a DENY PBR6 related to it, the session is destroyed.

You must apply this procedure after every modification you make to any PBR6. For example, you must follow this procedure after disabling a PBR6.

Note: PBR6s created on the NetScaler appliance do not work until they are applied.

To apply PBR6s by using the NetScaler command line:

At the command prompt, type:

- **apply ns PBR6**

To apply PBR6s by using the NetScaler GUI:

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Apply.

Enabling or Disabling a PBR6

By default, the PBR6s are enabled. This means that when PBR6s are applied, the NetScaler appliance automatically compares outgoing IPv6 packets against the configured PBR6s. If a PBR6 is not required in the lookup table, but it needs to be retained in the configuration, it must be disabled before the PBR6s are applied. After the PBR6s are applied, the NetScaler does not compare incoming packets against disabled PBR6s.

To enable or disable a PBR6 by using the NetScaler command line:

At the command prompt, type one of the following commands:

- **enable ns pbr** <name>
- **disable ns pbr** <name>

To enable or disable a PBR6 by using the NetScaler GUI:

1. Navigate to System > Network > PBRs.
2. On the PBR6s tab, select the PBR6, in the Action list, select Enable or Disable.

Renumbering PBR6s

You can automatically renumber the PBR6s to set their priorities to multiples of 10.

To renumber PBR6s by using the NetScaler command line:

At the command prompt, type:

- **renumber ns pbr6**

To renumber PBR6s by using the NetScaler GUI:

Navigate to System > Network > PBRs, on the PBR6s tab, in the Action list, select Renumber Priority (s).

MAC Address Wildcard Mask for PBRs

September 21, 2020

A wildcard mask parameter has been introduced for extended PBRs and PBR6s and is used with the source MAC address parameter to define a range of MAC addresses to be match against the source MAC address of outgoing packets.

Wild card masks specify which hexadecimal digits of the MAC address are used and which hexadecimal digits are ignored. The wildcard mask parameter specifies a series of ones and zeroes and has a length of 12 digits. Each digit is a mask for the corresponding hexadecimal digit of the MAC address. A zero digit in the wildcard mask indicates that the corresponding hexadecimal digit of the MAC address must be considered and a one digit indicates that the corresponding hexadecimal digit to be ignored.

The wildcard mask should meet the following conditions:

- Has only one series of zeroes
- Has only one series of ones
- Start with a series of zeroes

The following are some of the examples of valid wildcard masks:

- 000000111111
- 000000011111
- 000011111111

The following are some of the examples of invalid wildcard masks:

- 000000111100
- 111110000000
- 010101010101

For an PBR rule, a wildcard mask of 000000111111 for MAC address 96:fa:95:1d:67:4a defines the MAC address range 96:FA:95:00:00:00 - 96:FA:95:FF:FF:FF. This MAC address range is matched against the source MAC address of the outgoing packets.

To specify a range of source MAC addresses in a PBR rule by using the NetScaler command line:

At the command prompt, type:

- **add ns pbr** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show ns pbr** <pbrname>

Example:

```
> add ns pbr PBR-1 ALLOW -srcip 192.0.2.34 -srcMac 96:fa:95:1d:67:4a -srcMacMask 000000111111 -nexthop 198.51.100.1

 Done
```

To specify a range of source MAC addresses in an PBR6 rule by using the NetScaler command line:

At the command prompt, type:

- **add ns pbr6** <name> <action> -**srcMac** <mac_addr> -**srcMacMask** <string>
- **show pbr6** <pbr6name>

Example:

```
> add ns pbr6 PBR6-1 ALLOW  - srcipv6 2001:db8:0::7 -srcMac 96:fa:95:1d:67:4a   - srcMacMask 000000001111 -nexthop 2001:db8:0::1
```

```
Done
```

Using NULL Policy Based Routes to Drop Outgoing Packets

September 21, 2020

Some situations might demand that the NetScaler appliance drops specific outgoing packets instead of routing them, for example, in testing cases and during deployment migration.

NULL policy based routes can be used to drop specific outgoing packets. A NULL PBR is a type of PBR that has the nexthop parameter set to NULL. The NetScaler appliance drops outgoing packets that match a NULL PBR.

Configuring NULL PBRs for IPv4 Packets

To create a NULL PBR by using the NetScaler command line:

At the command prompt, type:

- **add ns pbr** <name> ALLOW [**-td** <positive_integer>] [-**srcIP** [<operator>] <srcIPVal>] [-**srcPort** [<operator>] <srcPortVal>] [-**destIP** [<operator>] <destIPVal>] [-**destPort** [<operator>] <destPortVal>] (-**nextHop NULL**) [**srcMac** <mac_addr> [-**srcMacMask** <string>]] [-**protocol** <protocol> | -**protocolNumber** <positive_integer>] [-**vlan** <positive_integer> | -**vxlan** <positive_integer>] [-**interface** <interface_name>] [-**priority** <positive_integer>] [-**msr** (**ENABLED** | **DISABLED**) [-**monitor** <string>]] [-**state** (**ENABLED** | **DISABLED**)[-**ownerGroup** <string>]
- **apply ns pbrs**
- **show ns pbr**<id>

To configure a NULL PBR by using the NetScaler GUI:

Navigate to **System** > **Network**> **PBRs**, on the **PBRs** tab, add a **new NULL PBR**, or edit an existing NULL PBR.

Sample configuration

In the following sample configuration, NULL PBR6 PBR6-NULL-EXAMPLE-1 is configured for dropping any outgoing IPv6 packets from interface 1/5.

```
> add ns pbr PBR6-NULL-EXAMPLE-1 ALLOW  - nextHop NULL -interface 1/5
Done
```

```

> apply ns pbr6
 Done
```

Troubleshooting Routing Issues

September 21, 2020

To make your troubleshooting process as efficient as possible, begin by gathering information about your network. You need to obtain the following information about the NetScaler appliance and other systems in the Network:

- Complete Topology diagram, including interface connectivity and intermediate switch details.
- Running Configuration. You can use the show running command to get the running configuration for ns.conf and ZebOS.conf.
- Output of the History command, to determine whether any configuration changes were made when the issue arose.
- Output of the Top and ps -ax commands, to determine whether any routing daemon is over utilizing the CPU or is misbehaving.
- Any routing related core files in /var/core - nsm, bgpd, ospfd, or ripd. Check the time stamp to see if they are relevant.
- dr_error.log and dr_info.log files from /var/log.
- Output of the date command and time details for all relevant systems. Print dates across all devices one after another, so that the times on the log messages can be correlated with various events.
- Relevant ns.log, newnslog files.
- Configuration files, log files and command history details from upstream and downstream routers.

Generic Routing FAQs

September 21, 2020

Users typically have the following questions about how to troubleshoot generic routing issues:

- How do I save the config files?

  The write command from VTYSH saves only ZebOS.conf. Run the save ns config command from NetScaler CLI to save both ns.conf and ZebOS.conf files.

- If I have configured both a static default route and a dynamically learned default route, which is the preferred default route?

  The dynamically learned route is the preferred default route. This behavior is unique to default routes. However, in case of the Network Services Module (NSM), unless the administrative distances are modified, a statically configured route in the RIB is preferred over a dynamic route. The route that is downloaded to the NSM FIB is the static route.

- How do I block the advertisement of default routes?

  The default route is not injected into ZebOS.

- How do I view the debug output of networking daemons?

  You can write debugging output from networking daemons to a file by entering the following log file command from the global configuration view in VTYSH:

  ```
  ns(config)# log file /var/ZebOS.log
  <!--NeedCopy-->
  ```

  You can direct debug output to the console by entering the terminal monitor command from VTYSH user view:

  ```
  ns# terminal monitor
  <!--NeedCopy-->
  ```

- How do I collect cores of running daemons?

  You can use the gcore utility to collect cores of running daemons for processing by gdb. This might be helpful in debugging misbehaving daemons without bringing the whole routing operation to a standstill.

  ```
  gcore [-s] [-c core] [executable] pid
  <!--NeedCopy-->
  ```

  The -s option temporarily stops the daemon while gathering the core image. This is a recommended option, because it guarantees that the resulting image shows the core in a consistent state.

  ```
  root@ns#gcore -s -c nsm.core /netscaler/nsm 342
  <!--NeedCopy-->
  ```

- How do I run a batch of ZebOS commands?

  You can run a batch of ZebOS commands from a file by entering the VTYSH -f <file-name> command. This does not replace the running configuration, but appends to it. However, by including commands to delete the existing configuration in the batch file and then add those for the new, desired configuration, you can use this mechanism to replace a specific configuration:

```
!
router bgp 234
network 1.1.1.1 255.255.255.0
!
route-map bgp-out2 permit 10
 set metric 9900
 set community 8602:300
!
<!--NeedCopy-->
```

Troubleshooting OSPF-Specific Issues

September 21, 2020

Before you start debugging any OSPF specific issue, you must collect information from the NetScaler appliance and all systems in the affected LAN, including upstream and downstream routers. To begin, enter the following commands:

1. show interface from both nscli and VTYSH
2. show ip ospf interface
3. show ip ospf neighbor detail
4. show ip route
5. show ip ospf route
6. show ip ospf database summary
    - If there are only few LSAs in the database, then enter show ip ospf database router, show ip ospf database A. network, show ip ospf database external, and other commands to get the full details of LSAs.
    - If there are a large number of LSAs in the database, enter the show ip ospf database self-originated command.
7. show ip ospf
8. show ns ip. This ensures that the details of all VIPs of interest are included.
9. Get the logs from peering devices and run the following command:

    ```
    gcore -s -c xyz.core /netscaler/ospfd <pid>
    ```

    Note: The gcore command is non-disruptive.

Collect additional information from the NetScaler as follows:

1. Enable logging of error messages by entering the following command from the global configuration view in VTYSH:

   ```
   ns(config)# log file /var/ospf.log
   <!--NeedCopy-->
   ```

2. Enable debugging ospf events and log them by using the following command:

   ```
   ns(config) #log file /var/ospf.log
   <!--NeedCopy-->
   ```

   Enable debug ospf lsa packet only if the number of LSAs in the database is relatively small (< 500).

Internet Protocol version 6 (IPv6)

September 21, 2020

A NetScaler appliance supports both server-side and client-side IPv6 and can therefore function as an IPv6 node. It can accept connections from IPv6 nodes (both hosts and routers) and from IPv4 nodes, and can perform Protocol Translation (RFC 2765) before sending traffic to the services. You have to license the IPv6 feature before you can implement it.

The following table lists some of the IPv6 features that the NetScaler appliance supports.

Table 1. Some Supported IPv6 Features

IPv6 features
IPv6 addresses for SNIPs (NSIP6, VIP6, and SNIP6)
Neighbor Discovery (Address Resolution, Duplicated Address Detection, Neighbor Unreachability Detection, Router Discovery)
Management Applications (ping6, telnet6, ssh6)
Static Routing and Dynamic routing (OSPF, BGP, RIPng, and ISIS)
Port Based VLANs
Access Control Lists for IPv6 addresses (ACL6)
IPv6 Protocols (TCP6, UDP6, ICMP6)
Server Side Support (IPv6 addresses for vservers, services)

IPv6 features
USIP (Use source IP) and DSR (Direct Server Return) for IPv6
SNMP and CVPN for IPv6
HA with native IPv6 node address
IPv6 addresses for MIPs
Path-MTU discovery for IPv6

Implementing IPv6 Support

IPv6 support is a licensed feature, which you have to enable before you can use or configure it. If IPv6 is disabled, the NetScaler does not process IPv6 packets. It displays the following warning when you run an unsupported command:

```
"Warning: Feature(s) not enabled [IPv6PT]"
<!--NeedCopy-->
```

The following message appears if you attempt to run IPv6 commands without the appropriate license:

```
"ERROR: Feature(s) not licensed"
<!--NeedCopy-->
```

After licensing the feature, use either of the following procedures to enable or disable IPv6.

NetScaler command line procedures

To enable or disable IPv6 by using the NetScaler command line:

At the command prompt, type one of the following commands:

- enable ns feature ipv6pt
- disable ns feature ipv6pt

NetScaler GUI procedures

To enable or disable IPv6 by using the NetScaler GUI:

1. Navigate to **System** > **Settings**, in the **Modes and Features** group, click **Configure Advanced Features**.
2. Select or clear the **IPv6 Protocol Translation** option.

VLAN Support

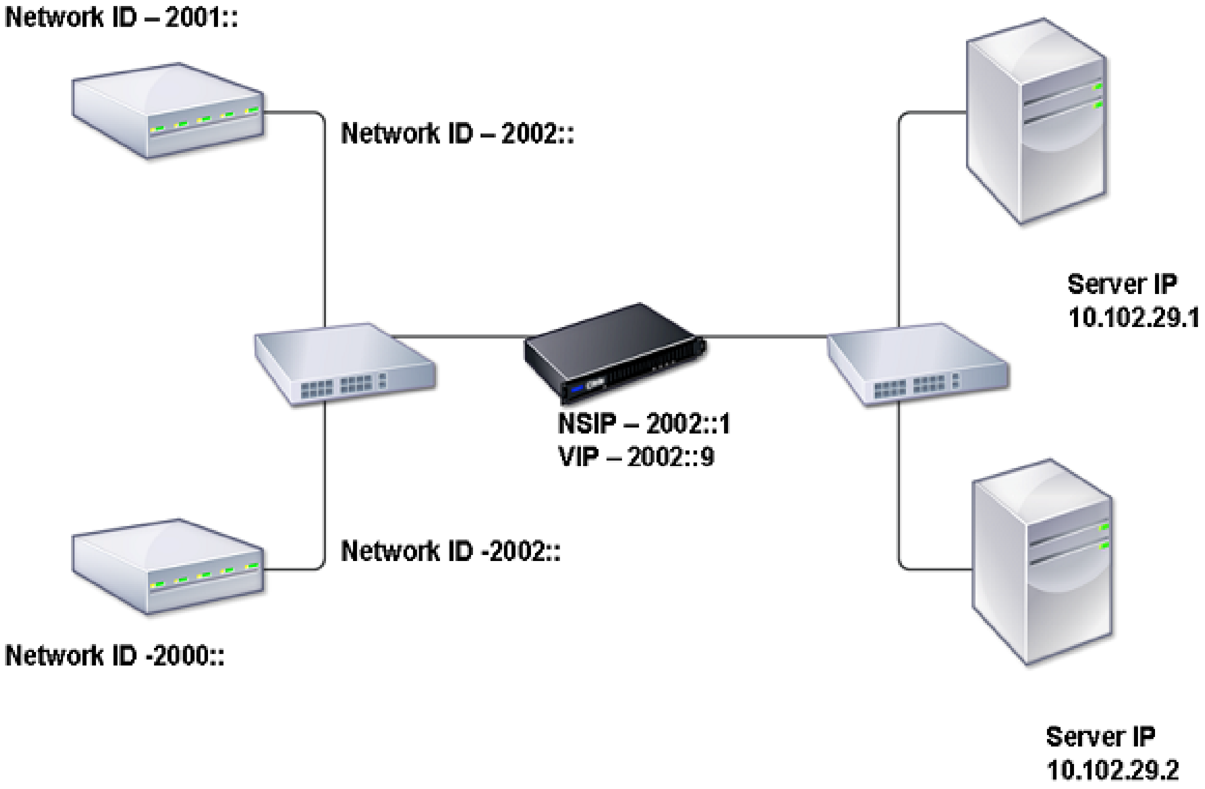
If you need to send broadcast or multicast packets without identifying the VLAN (for example, during DAD for NSIP, or ND6 for the next hop of the route), you can configure the NetScaler appliance to send the packet on all the interfaces with appropriate tagging. The VLAN is identified by ND6, and a data packet is sent only on the VLAN. For more information about ND6 and VLANs, see Configuring Neighbor Discovery.

Port-based VLANs are common for IPv4 and IPv6. Prefix-based VLANs are supported for IPv6.

Simple Deployment Scenario

Following is an example of a simple load balancing set-up consisting of an IPv6 vserver and IPv4 services, as illustrated in the following topology diagram.

Figure 1. IPv6 Sample Topology


Network ID – 2001::
Network ID – 2002::
Server IP
10.102.29.1
NSIP – 2002::1
VIP – 2002::9
Network ID -2002::
Network ID -2000::
Server IP
10.102.29.2

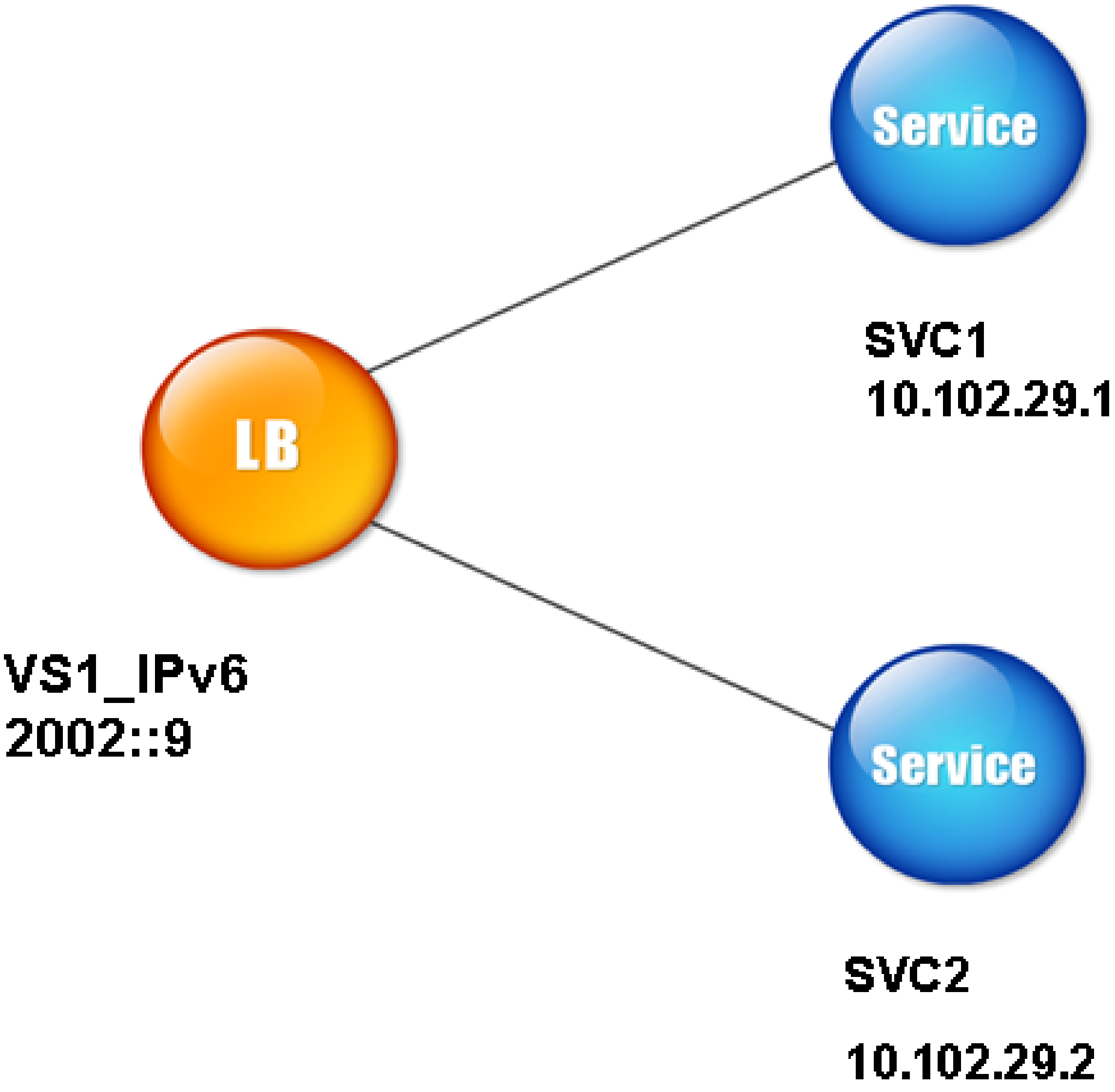
The following table summarizes the names and values of the entities that must be configured on the NetScaler.

Table 2. Sample Values for Creating Entities

Entity type	Name	Value
LB Vserver	VS1_IPv6	2002::9
Services	SVC1	10.102.29.1
	SVC2	10.102.29.2

The following figure shows the entities and values of the parameters to be configured on the NetScaler.

Figure 2. IPv6 Entity Diagram


Service
SVC1
10.102.29.1
LB
VS1_IPv6
2002::9
Service
SVC2
10.102.29.2

To configure this deployment scenario, you need to do the following:

1. Create an IPv6 service.
2. Create an IPv6 LB vserver.
3. Bind the services to the vserver.

NetScaler command line procedures

To create IPv4 services by using the NetScaler command line:

At the command prompt, type:

- **add service** <Name> <IPAddress> <Protocol> <Port>
- **sh service** <Name>

Example:

```
> add service SVC1 10.102.29.1 HTTP 80
Done

>add service SVC2 10.102.29.2 HTTP 80
Done
<!--NeedCopy-->
```

To create IPv6 vserver by using the NetScaler command line:

At the command prompt, type:

- **add lb vserver** <Name> <IPAddress> <Protocol> <Port>
- **sh lb vserver** <Name>

Example:

```
> add lb vserver VS1_IPv6 2002::9 HTTP 80
Done
<!--NeedCopy-->
```

To bind a service to an LB vserver by using the NetScaler command line:

At the command prompt, type:

- **bind lb vserver** <name> <service>
- **sh lb vserver** <name>

Example:

```
> bind lb vserver VS1_IPv6 SVC1
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To create IPv4 services by using the NetScaler GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Services**, click **Add**, and then set the following parameters:

- Service Name
- IP Address
- Protocol
- Port

To create IPv6 vserver by using the NetScaler GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, click **Add**, and select the **IPv6** check box.
2. Set the following parameters:
    - Name
    - Protocol
    - IP Address Type
    - IP Address
    - Port

To bind a service to an LB vserver by using the NetScaler GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. In the **Load Balancing Virtual Servers** page, select the vserver for which you want to bind the service (for example, VS1_IPv6).
3. Click **Open**.
4. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Services** tab, select the **Active** check box corresponding to the service that you want to bind to the vserver (for example, SVC1).
5. Click **OK**.
6. Repeat steps 1-4 to bind the service (for example, SVC2 to the vserver).

Host Header Modification

When an HTTP request has an IPv6 address in the host header, and the server does not understand the IPv6 address, you must map the IPv6 address to an IPv4 address. The IPv4 address is then used in the host header of the HTTP request sent to the vserver.

NetScaler command line procedures

To change the IPv6 address in the host header to an IPv4 address by using the NetScaler command line:

At the command prompt, type:

- **set ns ip6** <IPv6Address> -**map** <IPAddress>
- **sh ns ip6** <IPv6Address>

Example:

```
> set ns ip6 2002::9 -map 200.200.200.200
Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To change the IPv6 address in the host header to an IPv4 address by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs** and, on the **IPV6s** tab, select the IP address for which you want to configure a mapped IP address, for example, 2002:0:0:0:0:0:0:9, and click Edit.
2. In the **Mapped IP** text box, type the mapped IP address that you want to configure, for example, 200.200.200.200.

VIP Insertion

If an IPv6 address is sent to an IPv4-based server, the server may not understand the IP address in the HTTP header, and may generate an error. To avoid this, you can map an IPv4 address to the IPv6 VIP. Then, you can enable VIP insertion to enable insertion of the IPv4 VIP address and port number in the HTTP requests sent to the servers.

NetScaler command line procedures

To configure a map IPv6 address by using the NetScaler command line:

At the command prompt, type:

set ns ip6 <IPv6Address> -**map** <IPAddress>

Example:

```
> set ns ip6 2002::9 -map 200.200.200.200
 Done
<!--NeedCopy-->
```

To enable VIP insertion by using the NetScaler command line:

At the command prompt, type:

- **set lb vserver** <name> -**insertVserverIPPort** <Value>
- **sh lb vserver** <name>

Example:

```
> set lb vserver VS1_IPv6 -insertVserverIPPort ON
 Done

<!--NeedCopy-->
```

NetScaler GUI procedures

To configure a map IPv6 address by using the NetScaler GUI:

1. Navigate to **System** > **Network** > **IPs**, on the **IPV6s** tab, select the IP address for which you want to configure a map IP address, for example, 2002:0:0:0:0:0:0:9, and click **Edit**.
2. In the **Mapped IP** text box, type the map IP address that you want to configure, for example, 200.200.200.200.

To enable VIP insertion by using the NetScaler GUI:

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the virtual server that you want to enable port insertion, and click **Edit**.
2. In the **Advanced** tab, under **Traffic Settings**, in the **Vserver IP Port Insertion** drop-down list box, select **VIPADDR**.
3. In the **Vserver IP Port Insertion** text box, type the vip header.

Traffic Domains

September 21, 2020

> **Warning**
>
> Citrix recommends you to use Admin Partitions instead of using Traffic Domains. For more information, see Admin Partitioning page.

Traffic domains are a way to segment network traffic for different applications. You can use traffic domains to create multiple isolated environments within a NetScaler appliance. An application belonging to a specific traffic domain communicates with entities and processes traffic within that domain. The traffic belonging to one traffic domain cannot cross the boundary of another traffic domain.

Benefits of using Traffic Domains

The main benefits of using traffic domains on a NetScaler appliance are the following:

- **Use of duplicate IP addresses in a Network**. Traffic domains allow you to use duplicate IP address on the network. You can assign the same IP address or network address to multiple devices on a network, or multiple entities on a NetScaler appliance, as long as each of the duplicate address belongs to a different traffic domain.
- **Use of Duplicate entities on the NetScaler appliance**. Traffic domains also allow you to use duplicate NetScaler feature entities on the appliance. You can create entities with the same settings as long as each entity is assigned to a separate traffic domain.
  Note: Duplicate entities with same name is not supported.
- **Multitenancy**. Using traffic domains, you can provide hosting services for multiple customers by isolating each customer's type of application traffic within a defined address space on the network.

A traffic domain is uniquely identified by an identifier, which is an integer value. Each traffic domain needs a VLAN or a set of VLANs. The isolation functionality of the traffic domain depends on the VLANs bound to the traffic domain. More than one VLAN can be bound to a traffic domain, but the same VLAN cannot be a part of multiple traffic domains. Therefore, the maximum number of traffic domains that can be created depends on the number of VLANS configured on the appliance.

Default Traffic Domain

A NetScaler appliance has a preconfigured traffic domain, called the
default traffic domain, which has an ID of 0. All factory settings and configurations are part of the default traffic domain. You can create other traffic domains and then segment traffic between the default traffic domain and each of the other traffic domains. You cannot remove the default traffic domain from the NetScaler appliance. Any feature entity that you create without setting the traffic domain ID is automatically associated with the default traffic domain.

Note: Some features and configurations are supported only in the default traffic domain. They do not work in nondefault traffic domains. For a list of the features supported in all traffic domains, see Supported NetScaler Features in Traffic Domains.

How Traffic Domains Work

As an illustration of traffic domains, consider an example in which two traffic domains, with IDs 1 and 2, are configured on NetScaler appliance NS1.

In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the NetScaler appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. Servers S1 and S2 are connected to NS1 through L2 switch SW2-TD1.

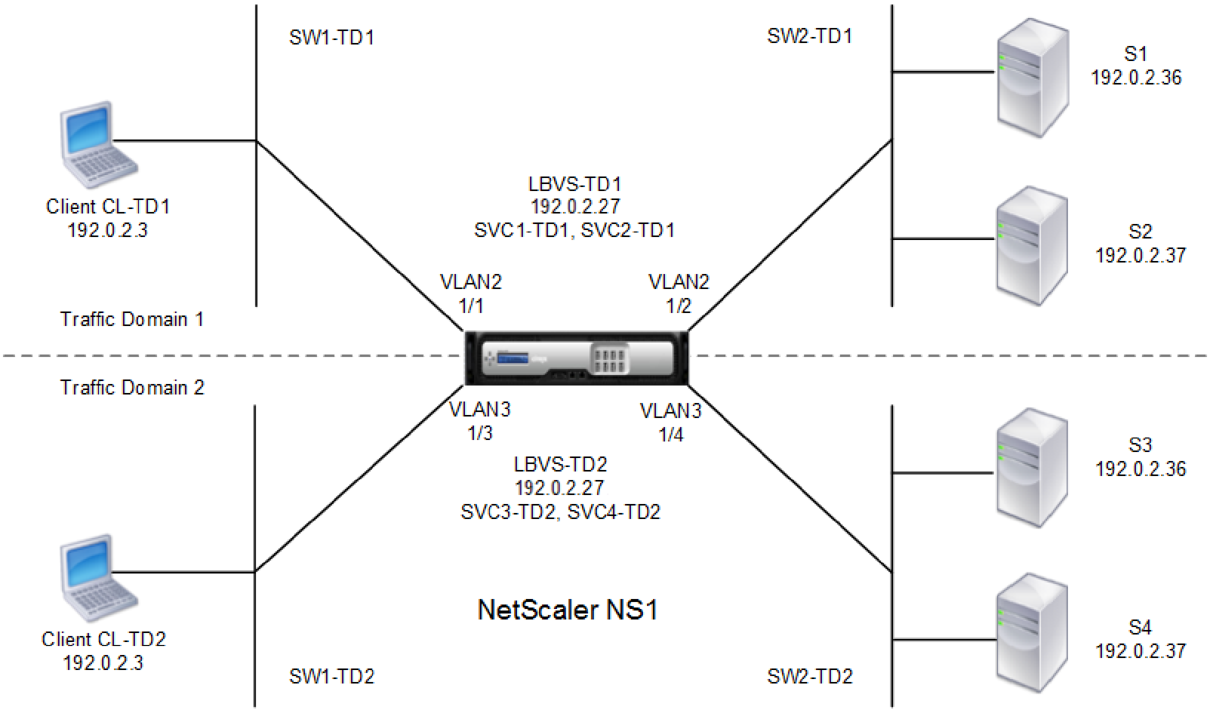
Client CL-TD1 is on a private network connected to NS1 through L2 switch SW1-TD1. SW1-TD1 and SW2-TD1 are connected to VLAN 2 of NS1. VLAN 2 is bound to traffic domain 1, which means that client CL-TD1 and servers S1 and S2 are part of traffic domain 1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the NetScaler appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. Servers S3 and S4 are connected to NS1 through L2 switch SW2-TD2. Client CL-TD2 is on a private network connected to NS1 through L2 switch SW1-TD2. SW1-TD2 and SW2-TD2 are connected to VLAN 3 of NS1. VLAN 3 is bound to traffic domain 2, which means that client CL-TD2 and servers S3 and S4 are part of traffic domain 2.

On the NetScaler appliance, entities LBVS-TD1 and LBVS-TD2 share the same settings, including the IP address. The same is true for SVC1-TD1 and SVC3-TD2, and for SVC2-TD1 and SVC4-TD2. This is possible because these entities are in different traffic domains.

Similarly, servers S1 and S3, S2 and S4 share the same IP address, and clients CL-TD1 and CL-TD2 each have the same IP address.

Figure 1. How traffic domains work


SW1-TD1
SW2-TD1
S1
192.0.2.36
Client CL-TD1
192.0.2.3
LBVS-TD1
192.0.2.27
SVC1-TD1, SVC2-TD1
S2
192.0.2.37
VLAN2
1/1
VLAN2
1/2
Traffic Domain 1
Traffic Domain 2
VLAN3
1/3
VLAN3
1/4
S3
192.0.2.36
LBVS-TD2
192.0.2.27
SVC3-TD2, SVC4-TD2
NetScaler NS1
Client CL-TD2
192.0.2.3
S4
192.0.2.37
SW1-TD2
SW2-TD2

The following table lists the settings used in the example.

Entity	Name	Details
Settings in traffic domain 1		

Entity	Name	Details
VLANs bound to traffic domain 1	VLAN 2	VLAN Id: 2 Interfaces bound: 1/1, 1/2
Client connected to TD1	CL-TD1 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD1	LBVS-TD1	IP address: 192.0.2.27
Service bound to virtual server LBVS-TD1	SVC1-TD1	IP address: 192.0.2.36
Service bound to virtual server LBVS-TD1	SVC2-TD1	IP address: 192.0.2.37
SNIP	SNIP-TD1 (for reference purposes only)	IP address: 192.0.2.27
Settings in traffic domain 2		
VLAN bound to traffic domain 2	VLAN 3	VLAN Id: 3 Interfaces bound: 1/3, 1/4
Client connected to TD2	CL-TD2 (for reference purposes only)	IP address: 192.0.2.3
Load balancing virtual server in TD2	LBVS-TD2	IP address: 192.0.2.27
Service bound to virtual server LBVS-TD2	SVC3-TD2	IP address: 192.0.2.36
Service bound to virtual server LBVS-TD2	SVC4-TD2	IP address: 192.0.2.37
SNIP in TD2	SNIP-TD2 (for reference purposes only)	IP address: 192.0.2.29

Following is the traffic flow in traffic domain 1:

1. Client CL-TD1 broadcasts an ARP request for the IP address of 192.0.2.27 through L2 switch SW1-TD1.
2. The ARP request reaches NS1 on interface 1/1, which is bound to VLAN 2. Because VLAN 2 is bound to traffic domain 1, NS1 updates the ARP table of traffic domain 1 for the IP address of client CL-TD1.
3. Because the ARP request is received on traffic domain 1, NS1 looks for an entity configured on traffic domain 1 that has an IP address of 192.0.2.27. NS1 finds that a load balancing virtual

server LBVS-TD1 is configured on traffic domain 1 and has the IP address 192.0.2.27.

4. NS1 sends an ARP response with the MAC address of interface 1/1.
5. The ARP reply reaches CL-TD1. CL-TD1 updates its ARP table for the IP address of LBVS-TD1 with the MAC address of interface 1/1 of NS1.
6. Client CL-TD1 sends a request to 192.0.2.27. The request is received by LBVS-TD1 on port 1/1 of NS1.
7. LBVS-TD1's load balancing algorithm selects server S2, and NS1 opens a connection between a SNIP in traffic domain 1 (192.0.2.27) and S2.
8. S2 replies to SNIP 192.0.2.27 on NS1.
9. NS1 sends S2's reply to client CL-TD1.

Following is the traffic flow in traffic domain 2:

1. Client CL-TD2 broadcasts an ARP request for the IP address of 192.0.2.27 through L2 switch SW1-TD2.
2. The ARP request reaches NS1 on interface 1/3, which is bound to VLAN 3. Because VLAN 3 is bound to traffic domain 2, NS1 updates traffic-domain 2's ARP-table entry for the IP address of client CL-TD2, even though an ARP entry for the same IP address (CL-TD1) is already present in the ARP table of traffic domain 1.
3. Because the ARP request is received in traffic domain 2, NS1 searches traffic domain 2 for an entity that has an IP address of 192.0.2.27. NS1 finds that load balancing virtual server LBVS-TD2 is configured in traffic domain 2 and has the IP address 192.0.2.27. NS1 ignores LBVS-TD1 in traffic domain 1, even though it has the same IP address as LBVS-TD2.
4. NS1 sends an ARP response with the MAC address of interface 1/3.
5. The ARP reply reaches CL-TD2. CL-TD2 updates its ARP table entry for the IP address of LBVS-TD2 with the MAC address of interface 1/3 of NS1.
6. Client CL-TD2 sends a request to 192.0.2.27. The request is received by LBVS-TD2 on interface 1/3 of NS1.
7. LBVS-TD2's load balancing algorithm selects server S3, and NS1 opens a connection between a SNIP in traffic domain 2 (192.0.2.29) and S3.
8. S2 replies to SNIP 192.0.2.29 on NS1.
9. NS1 sends S2's reply to client CL-TD2.

Supported NetScaler Features in Traffic Domains

The NetScaler features in the following list are supported in all traffic domains.

> **Important**
>
> Any NetScaler feature not listed below is supported only in the default traffic domain.

- ARP table
- ND6 table

- Bridge table
- All types of IPv4 and IPv6 addresses
- IPv4 and IPv6 routes
- ACL and ACL6
- PBR & PBR6
- INAT
- RNAT
- RNAT6
- MSR
- MSR6
- Net profiles
- SNMP MIBs
- Fragmentation
- Monitors (Scriptable monitors are not supported)
- Content Switching
- Cache Redirection
- Persistency
- Service (Domain based services are not supported)
- Servicegroup (Domain based service groups are not supported)
- Policies (*)
- PING
- TRACEROUTE
- PMTU
- High Availability (connection mirroring is not supported)
- Cluster (Supported on L2 clusters. Not supported on L3 clusters)
- Cookie Persistency
- MSS
- Logging (Syslog is not supported)
- Priority Queuing
- Surge Protection
- HTTP DOSP (**)
- Load balancing (The following types are not supported:)
    - TFTP
    - RTSP
    - Diameter
    - SIP
    - SMPP
- NAT46
- NAT64

- DNS64
- Forwarding Session Rules
- SNMP

> **Note**
>
> - * Policies do not have global binding points for traffic domains. However, policies can be bound to a specific load balancing virtual server of a traffic domain.
> - ** HTTP DOSP policies do not have global binding points for traffic domains. However, HTTP DOSP policies can be bound to a specific load balancing service of a traffic domain.
> - Global Server Loading Balancing (GSLB) and ADNS features in NetScaler are not aware of Traffic Domains. If the GSLB configuration needs to be shared across all traffic domains then GSLB methods Static Proximity and Round Trip Time (RTT) do not work. As a workaround in this scenario you can use GSLB methods other than RTT and Static Proximity. For more information, see http://support.citrix.com/article/CTX202277.

Configuring Traffic Domains

Configuring a traffic domain on the NetScaler appliance consists of the following tasks:

- **Add VLANs.** Create VLANs and bind specified interfaces to them.
- **Create a traffic domain entity and bind VLANs to it.** This involves the following two tasks:
  - Create a traffic domain entity uniquely identified by an ID, which is an integer value.
  - Bind the specified VLANs to the traffic domain entity. All the interfaces that are bound to the specified VLANs are associated with the traffic domain. More than one VLAN can be bound to a traffic domain, but a VLAN cannot be a part of multiple traffic domains.
- **Create feature entities on the traffic domain**. Create the required feature entities in the traffic domain. The CLI commands and configuration dialog boxes of all the supported features in a nondefault traffic domain include a parameter called a *traffic domain identifier* (td). When configuring a feature entity, if you want the entity to be associated with a particular traffic domain, you must specify the td. Any feature entity that you create without setting the td is automatically associated with the default traffic domain.

To give you an idea of how feature entities are associated with a traffic domain, this topic covers the procedures for configuring all the entities mentioned in the figure titled How Traffic Domains Work.

The NetScaler command line has two commands for these two tasks, but the NetScaler GUI combines them in a single dialog box.

NetScaler command line procedures

To create a VLAN and bind interfaces to it by using the NetScaler command line:

At the command prompt, type:

- **add vlan** <id>
- **bind vlan** <id> -**ifnum** <slot/port>
- **show vlan** <id>

To create a traffic domain entity and bind VLANs to it by using the NetScaler command line:

At the command prompt, type:

- **add ns trafficdomain** <td>
- **bind ns trafficdomain** <td> -**vlan** <id>
- **show ns trafficdomain** <td>

To create a service by using the NetScaler command line:

At the command prompt, type:

- **add service** <name> <IP> <serviceType> <port> -**td** <id>
- **show service** <name>

To create a load balancing virtual server and bind services to it by using the NetScaler command line:

At the command prompt, type:

- **add lb vserver** <name> <serviceType> <IPAddress> <port> -**td** <id>
- **bind lb vserver** <name> <serviceName>
- **show lb vserver** <name>

NetScaler GUI procedures

To create a VLAN by using the NetScaler GUI:

Navigate to **System** > **Network** > **VLANs**, click **Add**, and set the parameters.

To create a traffic domain entity by using the NetScaler GUI:

Navigate to **System** > **Network** > **Traffic Domains**, click **Add**, and in the **Create Traffic Domain** dialog box, set the parameters.

To create a service by using the NetScaler GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Services**, click **Add**, and set the parameters.

To create a load balancing virtual server by using the NetScaler GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, click **Add**, and set the parameters.

Inter Traffic Domain Entity Bindings

September 21, 2020

You can bind services in one traffic domain to a virtual server in another traffic domain. All the services to be bound to a virtual server in a different traffic domain must reside in the same traffic domain.

You configure this support by using the existing bind lb vserver command or the related NetScaler GUI procedure.

This capability can facilitate interaction between different traffic domains. In an enterprise, servers can be grouped in different traffic domains. Virtual servers are created in a traffic domain that faces the internet. A virtual server from this traffic domain can be configured to load balance servers in another traffic domain. This virtual server receives connection requests from the Internet to be forwarded to the bound servers.

When a NetScaler appliance is used in a cloud infrastructure, each tenant can be assigned a separate traffic domain, and all the resources (including servers) for a tenant can be grouped together in the tenant's traffic domain. For each tenant, a virtual server is created for load balancing servers in its traffic domain. All of these virtual servers are grouped together in a single traffic domain that faces the Internet.

Consider an example of in which cloud service provider Example-Cloud-A has three traffic domains, with IDs 10, 20, and 30, configured on NetScaler appliance NS1.

Example-Org-A and Example-Org-B are tenants of Example-Cloud-A. Tenant A is assigned traffic domain 20, and tenant B is assigned domain 30. Servers S1 and S2 reside in traffic domain 20 and servers S3 and S4 reside in traffic domain 30.

Traffic domain 10 faces the internet. Virtual servers LBVS-1 and LBVS-2 are created in traffic domain 10. LBVS-1, in traffic domain 10, is configured to load balance servers S1 and S2, which are in traffic domain 20. LBVS-2, in traffic domain 10, is configured to load balance servers S3 and S4, which are in traffic domain 30.

Therefore, these virtual servers accept Internet connection requests for servers that are in a different traffic domain than that of the virtual servers.

virtual MAC Based Traffic Domains

September 21, 2020

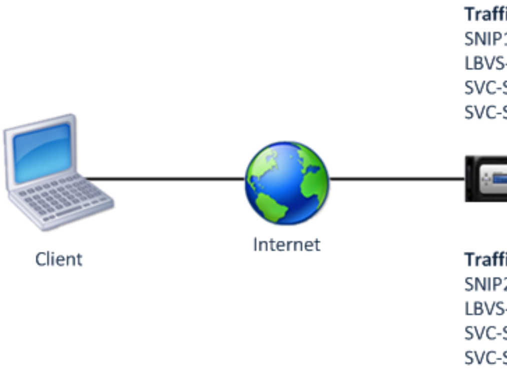
You can associate a traffic domain with a virtual MAC address instead of with VLANs. The NetScaler appliance then sends the traffic domain's virtual MAC address in all responses to ARP queries for network entities in that domain. As a result, the appliance can segregate subsequent incoming traffic for different traffic domains on the basis of the destination MAC address, because the destination MAC address is the virtual MAC address of a traffic domain. After creating entities on a traffic domain, you can easily manage and monitor them by performing traffic domain level operations.

Consider an example in which two traffic domains, with IDs 1 and 2, are configured on NetScaler appliance NS1. The NetScaler creates a virtual MAC address VMAC1 and associates it with traffic domain 1. Similarly, the NetScaler created another virtual MAC address VMAC2 and associates with traffic domain 2.

In traffic domain 1, load balancing virtual server LBVS-TD1 is configured to load balance traffic across servers S1 and S2. On the NetScaler appliance, servers S1 and S2 are represented by services SVC1-TD1 and SVC2-TD1, respectively. A subnet IP address (SNIP) SNIP1 is configured for enabling the NetScaler to communicate with S1 and S2. Because VMAC1 is associated with traffic domain 1, the NetScaler sends VMAC1 as the MAC address in all ARP announcements and ARP responses for LBVS-TD1 and SNIP1.

Similarly in traffic domain 2, load balancing virtual server LBVS-TD2 is configured to load balance traffic across S3 and S4. On the NetScaler appliance, servers S3 and S4 are represented by services SVC3-TD2 and SVC4-TD2, respectively. A SNIP address SNIP2 is configured for enabling the NetScaler to communicate with S3 and S4. Because VMAC2 is associated with traffic domain 2, the NetScaler sends VMAC2 as the MAC address in all ARP announcements and ARP responses for LBVS-TD2 and SNIP2.

The NetScaler segregate subsequent incoming traffic for traffic domains 1 or 2 on the basis of the destination MAC address, if the destination MAC address is VMAC1 or VMAC2.



The following table lists the settings used in the example: Vitrual MAC based traffic domain example settings.

Before you Begin

Following are points to consider before you configure virtual MAC based traffic domain:

1. virtual MAC based traffic domains are easiest way to achieve network traffic segregation.
2. Because virtual MAC based traffic domains segregate network traffic based on virtual MAC addresses and not VLANS, you cannot create duplicate IP addresses on different virtual MAC based traffic domains on a NetScaler appliance.
3. virtual MAC based traffic domains do not work when the NetScaler is deployed only in L2 Mode.
4. Both VLAN and virtual MAC based traffic domains can coexist on a NetScaler appliance. virtual MAC based traffic domains actually runs on all VLANs that are not bound to any VLAN based traffic domain.

Configuration Steps

Configuring a virtual MAC based traffic domain on a NetScaler appliance consists of the following tasks:

- Create a traffic domain entity and enable the virtual MAC option. Create a traffic domain entity uniquely identified by an ID, which is an integer value, and then enable the virtual MAC option. After creating the traffic domain entity, the NetScaler appliance creates a virtual MAC address and then associates it to the traffic domain entity.
- Create feature entities on the traffic domain. Create the required feature entities in the traffic domain by specifying the traffic domain identifier (td) when configuring these feature entities. NetScaler owned network entities created in a virtual MAC based traffic domain are associated with the virtual MAC address, which is associated with the traffic domain. The NetScaler appliance then sends the traffic domain's virtual MAC address in ARP announcements and ARP responses for these network entities.

NetScaler command line procedures

To create a virtual MAC based traffic domain by using the NetScaler command line:

At the command prompt, type:

```
add ns trafficDomain <td> [-vmac ( ENABLED      DISABLED )]
```

-
- show ns trafficdomain <td>

To configure a SNIP address by using the NetScaler command line:

At the command prompt, type:

- add ns ip <IPAddress> <netmask> -type SNIP –td <id>
- show ns ip <IPAddress> -td <id>

To create a service by using the NetScaler command line:

At the command prompt, type:

- add service <name> <IP> <serviceType> <port> -td <id>
- show service <name> -td <id>

To create a load balancing virtual server and bind services to it by using the NetScaler command line:

At the command prompt, type:

- add lb vserver <name> <serviceType> <IPAddress> <port> -td <id>
- bind lb vserver <name> <serviceName>
- show lb vserver <name> -td <id>

Example:

```
> add ns trafficDomain 1 -vmac ENABLED
 Done
> add ns trafficDomain 2 -vmac ENABLED
 Done

> add ns ip 192.0.2.5 255.255.255.0 -type -SNIP -td 1
 Done
> add service  SVC-S1-TD1  192.0.2.10 HTTP 80 -td 1
 Done
> add service  SVC-S2-TD1  192.0.2.20 HTTP 80 -td 1
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.15 80 -td 1
 Done
> bind lb vserver  LBVS-TD1 SVC-S1-TD1
 Done
> bind lb vserver  LBVS-TD1 SVC-S2-TD1
 Done

> add ns ip 192.0.2.6 255.255.255.0 -type -SNIP -td 2
 Done
> add service  SVC-S3-TD2  192.0.2.30 HTTP 80 -td 2
 Done
> add service  SVC-S4-TD2  192.0.2.40 HTTP 80 -td 2
 Done
> add lb vserver  LBVS-TD1 HTTP   203.0.113.16 80 -td 1
 Done
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
```

```
> bind lb vserver  LBVS-TD2 SVC-S3-TD2
 Done
<!--NeedCopy-->
```

NetScaler GUI procedures

To create a virtual MAC based traffic domain by using the NetScaler GUI:

1. Navigate to System > Network > Interfaces.
2. In the details pane, click Add.
3. On the Create Traffic Domain page, set the following parameters:
    - Traffic Domain ID*
    - Enable Mac
4. Click Create.

To configure a SNIP address by using the NetScaler GUI:

1. Navigate to System > Network > IPs > IPv4
2. Navigate to Network > IPs > IPv4
3. In the details pane, click Add
4. In the Create IP page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
    - IP Address
    - Netmask
    - IP Type
    - Traffic Domain ID
5. Click Create.

To create a service by using the NetScaler GUI:

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, click Add.
3. In the Basic Settings Page, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
    - Service Name
    - Server
    - Protocol
    - Port
    - Traffic Domain ID
4. Click Continue, and click Done.
5. Repeat steps 2-4 to create another service.
6. Click Close.

To create a load balancing virtual server and bind services to it by using the NetScaler GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the Load Balancing Virtual Servers pane, click Add.
3. In the Create Virtual Servers (Load Balancing) dialog box, set the following parameters. For a description of a parameter, hover the mouse cursor over the corresponding field.
   - Name
   - IP Address
   - Protocol
   - Port
   - Traffic Domain ID
4. Click Continue, on the Service Pane, click >.
5. On the Service page, click Insert, and then select the check box for the services that you want to bind to the virtual server.
6. Click Continue, and click Done.
7. Repeat steps 2-5 to create another virtual server

VXLAN

September 21, 2020

NetScaler appliances support Virtual eXtensible Local Area Networks (VXLANs). A VXLAN overlays Layer 2 networks onto a layer 3 infrastructure by encapsulating Layer-2 frames in UDP packets. Each overlay network is known as a VXLAN Segment and is identified by a unique 24-bit identifier called the VXLAN Network Identifier (VNI). Only network devices within the same VXLAN can communicate with each other.

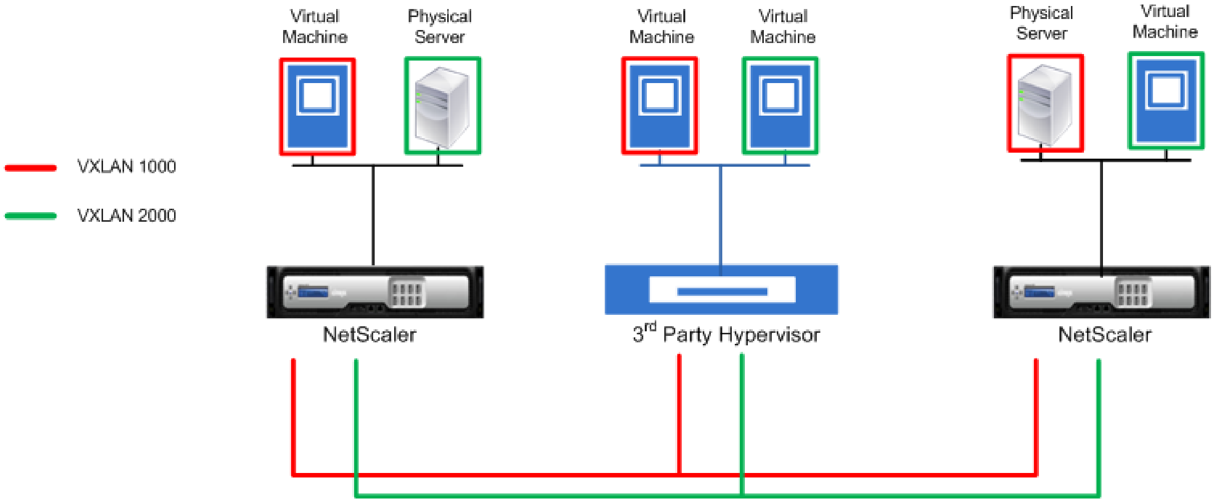
VXLANs provide the same Ethernet Layer 2 network services that VLANs do, but with greater extensibility and flexibility. The two main benefits of using VXLANs are the following:

- **Higher scalability.** Server virtualization and cloud computing architectures have dramatically increased the demand for isolated Layer 2 networks in a datacenter. The VLAN specification uses a 12-bit VLAN ID to identify a Layer 2 network, so you cannot scale beyond 4094 VLANs. That number can be inadequate when the requirement is for thousands of isolated Layer 2 networks. The 24-bit VNI accommodates up to 16 million VXLAN segments in the same administrative domain.
- **Higher flexibility.** Because VXLAN carries Layer 2 data frames over Layer 3 packets, VXLANs extend L2 networks across different parts of a datacenter and across geographically separated datacenters. Applications that are hosted in different parts of a datacenter and in different datacenters but are part of the same VXLAN appear as one contiguous network.

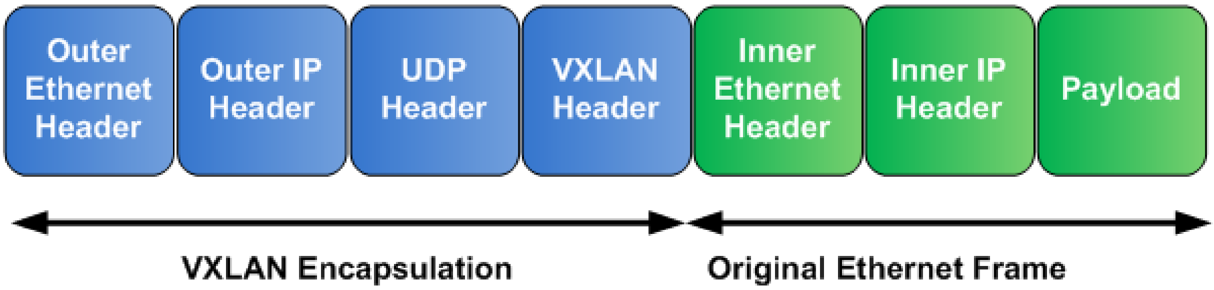
How VXLANs Work

VXLAN Segments are created between VXLAN Tunnel End Points (VTEPs). VTEPs support the VXLAN protocol and perform VXLAN encapsulation and decapsulation. You can think of a VXLAN segment as a tunnel between two VTEPs, where one VTEP encapsulates a Layer2 frame with a UDP header and an IP header and sends it through the tunnel. The other VTEP receives and decapsulates the packet to get the Layer 2 frame. A NetScaler appliance is one example of a VTEP. Other examples are third-party hypervisors, VXLAN aware virtual machines, and VXLAN capable switches.

The following illustration displays virtual machines and physical servers connected through VXLAN tunnels.


VXLAN 1000
VXLAN 2000
Virtual Machine
Physical Server
Virtual Machine
Virtual Machine
Physical Server
Virtual Machine
NetScaler
3rd Party Hypervisor
NetScaler

The following illustration displays the format of a VXLAN packet.


Outer Ethernet Header
Outer IP Header
UDP Header
VXLAN Header
Inner Ethernet Header
Inner IP Header
Payload
VXLAN Encapsulation
Original Ethernet Frame

VXLANs on a NetScaler appliance use a Layer 2 mechanism for sending broadcast, multicast, and unknown unicast frames. A VXLAN supports the following modes for sending these L2 frames.

- **Unicast mode**: In this mode, you specify the IP addresses of VTEPs while configuring a VXLAN on a NetScaler appliance. The NetScaler appliance sends broadcast, multicast, and unknown unicast frames over Layer 3 to all VTEPs of this VXLAN.
- **Multicast mode**: In this mode, you specify a multicast group IP address while configuring a VXLAN on a NetScaler appliance. NetScaler appliances do not support Internet Group Management Protocol (IGMP) protocol. NetScaler appliances rely on the upstream router to join a multicast group, which shares a common multicast group IP address. The NetScaler appliance sends

broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN.

Similar to a Layer 2 bridge table, NetScaler appliances maintain VXLAN mapping tables based on the inner and outer header of the received VXLAN packets. This table maps of remote host MAC addresses to VTEP IP addresses for a particular VXLAN. The NetScaler appliance uses the VXLAN mapping table to look up the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in the VXLAN table, the NetScaler appliance sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

Because VXLANs function similarly to VLANs, most of the NetScaler features that support VLAN as a classification parameter support VXLAN. These features include an optional VXLAN parameter setting, which specifies the VXLAN VNI.

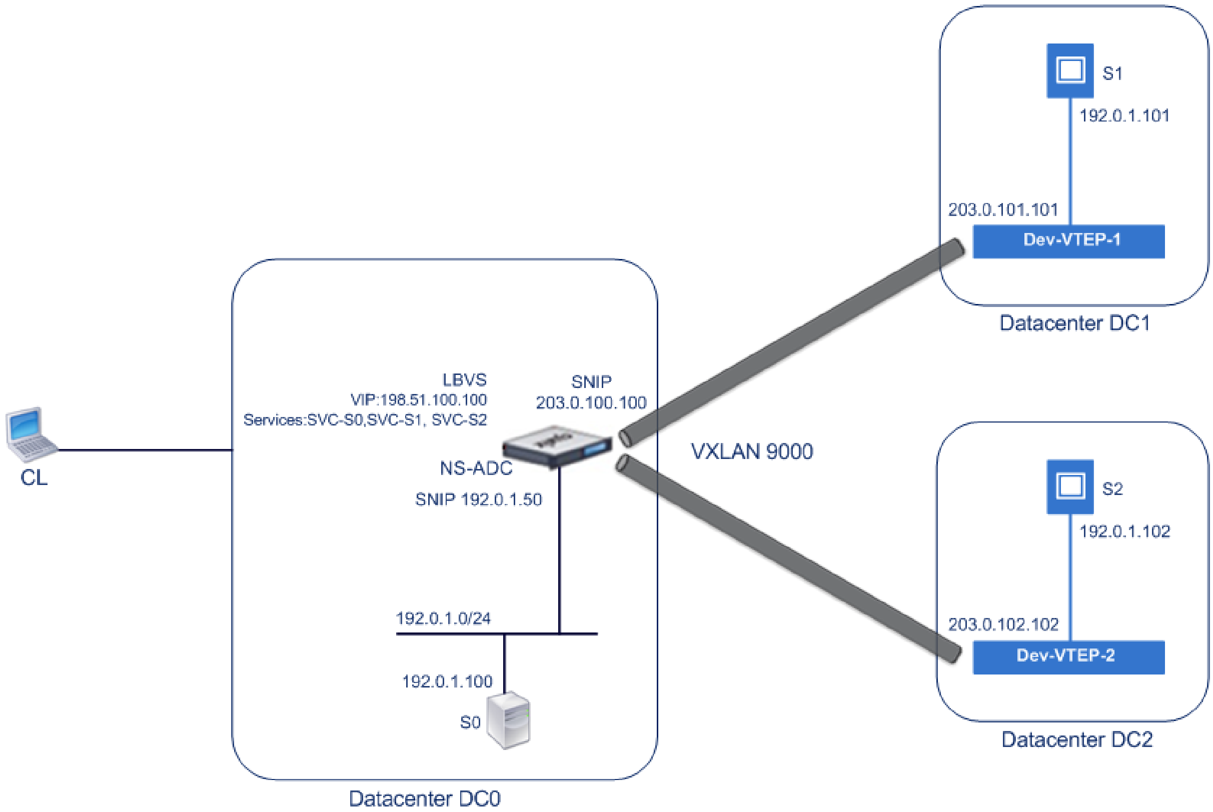
In a high availability (HA) configuration, the VXLAN configuration is propagated or synchronized to the secondary node.

VXLAN Use Case: Load Balancing across Datacenters

To understand the VXLAN functionality of a NetScaler appliance, consider an example in which Example Corp hosts a site at www.example.com. To ensure application availability, the site is hosted on three servers, S0, S1, and S2. A load balancing virtual server, LBVS, on NetScaler appliance NS-ADC is used to load balance these servers. S0, S1, and S2 reside in datacenters DC0, DC1, and DC2, respectively. In DC0, server S0 is connected to NS-ADC.

S0 is a physical server, and S1 and S2 are virtual machines (VMs). S1 runs on virtualization host device Dev-VTEP-1 in datacenter DC1, and S2 runs on host device Dev-VTEP-2 in DC2. NS-ADC, Dev-VTEP-1, and Dev-VTEP-2 support the VXLAN protocol.

S0, S1, and S2 are part of the same private subnet, 192.0.1.0/24. S0, S1, and S2 be part of a common broadcast domain, VXLAN 9000 is configured on NS-ADC, Dev-VTEP-1, and Dev-VTEP-2. Servers S1 and S2 are made part of VXLAN9000 on Dev-VTEP-1 and Dev-VTEP-2, respectively.


S1
192.0.1.101
203.0.101.101
Dev-VTEP-1
Datacenter DC1
LBVS
VIP:198.51.100.100
Services:SVC-S0,SVC-S1, SVC-S2
SNIP
203.0.100.100
NS-ADC
SNIP 192.0.1.50
CL
VXLAN 9000
S2
192.0.1.102
203.0.102.102
Dev-VTEP-2
Datacenter DC2
192.0.1.0/24
192.0.1.100
S0
Datacenter DC0

The following table lists the settings used in this example:
VXLAN settings.

Services SVC-S0, SVC-S1, and SVC-S2 on NS-ADC represent S0, S1, and S2. As soon as these services are configured, NS-ADC broadcasts ARP requests for S0, S1, and S2 to resolve IP-to-MAC mapping. These ARP requests are also sent over VXLAN 9000 to Dev-VTEP-1 and Dev-VTEP-2.

Following is the traffic flow for resolving the ARP request for S2:

1. NS-ADC broadcasts an ARP request for S2 to resolve IP-to-MAC mapping. This packet has:
   - Sourced IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
   - Source MAC address = MAC address of the NS-ADC's interface from which the packet is sent out = NS-MAC-1
2. NS-ADC prepares the ARP packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
   - VXLAN header with an ID (VNI) of 9000
   - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
3. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-1 and Dev-VTEP-2 on VXLAN-9000. The encapsulated packet has:
   - Source IP address = SNIP-VTEP-0 (203.0.100.100).
4. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN

header and learns the VXLAN ID of the packet. The resulting packet is the ARP request packet for S2, which is same as in step 1.

5. From the inner and outer header of the VXLAN packet, Dev-VTEP-2 makes an entry in its VXLAN mapping table that shows the mapping of MAC address (NS-MAC-1) and SNIP-VTEP-0 (203.0.100.100) for VXLAN9000.
6. Dev-VTEP-2 sends the ARP packet to S2. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000.
7. S2 responds with its MAC address (MAC-S2). The ARP response packet has:
   - Destination IP address = Subnet IP address SNIP-for-Servers (192.0.1.50)
   - Destination MAC address = NS-MAC-1
8. S2's response packet reaches Dev-VTEP-2. Dev-VTEP-2 performs a lookup in its VXLAN mapping table and gets a match for the destination MAC address NS-MAC-1. The Dev-VTEP-2 now knows that NS-MAC-1 is reachable through SNIP-VTEP-0 (203.0.100.100) over VXLAN 9000. Dev-VTEP-2 encapsulates the ARP response with VXLAN and UDP headers, and sends the resultant packet to SNIP-VTEP-0 (203.0.100.100) of NS-ADC.
9. NS-ADC on receiving the packet, decapsulates the packet by removing the VXLAN and UDP headers. The resultant packet is S2's ARP response. NS-ADC updates it VXLAN mapping table for S2's MAC address (MAC-S2) with Dev-VTEP-2's IP address (203.0.102.102) for VXLAN 9000. NS-ADC also updates it ARP table for S2's IP address (192.0.1.102) with S2's MAC address (MAC-S2).

Following is the traffic flow for load balancing virtual server LBVS in this example:

1. Client CL sends a request packet to LBVS of NS-ADC. The request packet has:
   - Source IP address = IP address of client CL (198.51.100.90)
   - Destination IP address = IP address (VIP) of LBVS = 198.51.110.100
2. LBVS of NS-ADC receives the request packet, and its load balancing algorithm selects server S2 of datacenter DC2.
3. NS-ADC processes the request packet, changing its destination IP address to the IP address of S2 and its source IP address to one of the Subnet IP (SNIP) addresses configured on NS-ADC. The request packet has:
   - Source IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
   - Destination IP address = IP address of S2 (192.0.1.102)
4. NS-ADC finds a VXLAN mapping entry for S2 in its bridge table. This entry indicates that S2 is reachable through Dev-VTEP-2 over VXLAN 9000.
5. NS-ADC prepares the packet to be sent over the VXLAN 9000 by encapsulating the packet with following headers:
   - VXLAN header with an ID (VNI) of 9000
   - Standard UDP header, UDP checksum set to 0×0000, and destination port set to 4789.
6. NS-ADC sends the resulting encapsulated packet to Dev-VTEP-2. The request packet has:

   - Source IP address = SNIP address = SNIP-VTEP-0 (203.0.100.100)
   - Destination IP address = IP address of Dev-VTEP-2 (203.0.102.102)
7. Dev-VTEP-2 receives the UDP packet and decapsulates the UDP header, from which Dev-VTEP-2 learns that the packet is a VXLAN related packet. Dev-VTEP-2 then decapsulates the VXLAN header and learns the VXLAN ID of the packet. The resulting packet is the same packet as in step 3.
8. Dev-VTEP-2 then forwards the packet to S2.
9. S2 processes the request packet and sends the response to the SNIP address of NS-ADC. The response packet has:
   - Source IP address = IP address of S2 (192.0.1.102)
   - Destination IP address = Subnet IP address on NS-ADC= SNIP-for-Servers (192.0.1.50)
10. Dev-VTEP-2 encapsulates the response packet in the same way that NS-ADC encapsulated the request packet in steps 4 and 5. Dev-VTEP-2 then sends the encapsulated UDP packet to SNIP address SNIP-for-Servers (192.0.1.50) of NS-ADC.
11. NS-ADC, upon receiving the encapsulated UDP packet, decapsulates the packet by removing the UDP and VXLAN headers in the same way that Dev-VTEP-2 decapsulated the packet in step 7. The resultant packet is the same response packet as in step 9.
12. NS-ADC then uses the session table for load balancing virtual server LBVS, and forwards the response packet to client CL. The response packet has:
   - Source IP address = IP address of client CL (198.51.100.90)
   - Destination IP address = IP address (VIP) of LBVS (198.51.110.100)

Points to Consider for Configuring VXLANs

Consider the following points before configuring VXLANs on a NetScaler appliance:

- A maximum of 2048 VXLANs can be configured on a NetScaler appliance.
- VXLANs are not supported in a cluster.
- Link-local IPv6 addresses cannot be configured for each VXLAN.
- NetScaler appliances do not support Internet Group Management Protocol (IGMP) protocol to form a multicast group. NetScaler appliances rely on the IGMP protocol of its upstream router to join a multicast group, which share a common multicast group IP address. You can specify a multicast group IP address while creating VXLAN bridge table entries but the multicast group must be configured on the upstream router. The NetScaler appliance sends broadcast, multicast, and unknown unicast frames over Layer 3 to the multicast group IP address of this VXLAN. The upstream router then forwards the packet to all the VTEPs that are part of the multicast group.
- VXLAN encapsulation adds an overhead of 50 bytes to each packet:

  Outer Ethernet Header (14) + UDP header (8) + IP header (20) + VXLAN header (8) = 50 bytes

To avoid fragmentation and performance degradation, you must adjust the MTU settings of all network devices in a VXLAN pathway, including the VXLAN VTEP devices, to handle the 50 bytes of overhead in the VXLAN packets.

Important: Jumbo frames are not supported on the NetScaler VPX virtual appliances, NetScaler SDX appliances, and NetScaler MPX 15000/17000 appliances. These appliances support an MTU size of only 1500 bytes and cannot be adjusted to handle the 50 bytes overhead of VXLAN packets. VXLAN traffic might be fragmented or suffer performance degradation, if one of these appliances is in the VXLAN pathway or acts as a VXLAN VTEP device.

- On NetScaler SDX appliances, VLAN filtering does not work for VXLAN packets.
- You cannot set a MTU value on a VXLAN.
- You cannot bind interfaces to a VXLAN.

Configuration Steps

Configuring a VXLAN on a NetScaler appliance consists of the following tasks.

- **Add a VXLAN entity**. Create a VXLAN entity uniquely identified by a positive integer, which is also called the VXLAN Network Identifier (VNI). In this step, you can also specify the destination UDP port of remote VTEP on which the VXLAN protocol is running. By default, the destination UDP port parameter is set to 4789 for the VXLAN entity. This UDP port setting must match the settings on all remote VTEPs for this VXLAN. You can also bind VLANs to this VXLAN. The traffic (which includes broadcasts, multicasts, unknown unicasts) of all bound VLANs are allowed over this VXLAN. If no VLANs are bound to the VXLAN, the NetScaler appliance allows traffic of all VLANs, on this VXLAN, that are not part of any other VXLANs.
- **Bind the local VTEP IP address and to the VXLAN entity**. Bind one of the configured SNIP address to the VXLAN to source outgoing VXLAN packets.
- **Add a bridgetable entry**. Add a bridgetable entry specifying the VXLAN ID and the remote VTEP IP address for the VXLAN to be created.
- **(Optional) Bind different feature entities to the configured VXLAN**. VXLANs function similarly to VLANs, most of the NetScaler appliance features that support VLAN as a classification parameter also support VXLAN. These features include an optional VXLAN parameter setting, which specifies the VXLAN VNI.
- **(Optional) Display the VXLAN mapping table**. Display the VXLAN mapping table, which includes mapping entries for remote host MAC address to VTEP IP address for a particular VXLAN. In other words, a VXLAN mapping states that a host is reachable through the VTEP on a particular VXLAN. The NetScaler appliance learns VXLAN mappings and updates its mapping table from the VXLAN packets it receives. The NetScaler appliance uses the VXLAN mapping table to lookup for the destination MAC address of a Layer 2 frame. If an entry for this MAC address is present in

the VXLAN table, the NetScaler appliance sends the Layer 2 frame over Layer 3, using the VXLAN protocol, to the mapped VTEP IP address specified in the mapping entry for a VXLAN.

NetScaler command line procedures

To add a VXLAN entity by using NetScaler command line:

At the command prompt, type

- **add vxlan** <id>
- **show vxlan**<id>

To bind the local VTEP IP address to the VXLAN by using NetScaler command line:

At the command prompt, type

- **bind vxlan** <id> -**SrcIP** <IPaddress>
- **show vxlan** <id>

To add a bridgetable by using NetScaler command line:

At the command prompt, type

- **add bridgetable** -**mac** <macaddress> -**vxlan** <ID> -**vtep** <IPaddress>
- **show bridgetable**

To display the VXLAN forwarding table by using the command line:

At the command prompt, type:

- **show bridgetable**

NetScaler GUI procedures

To add a VXLAN entity and bind a local VTEP IP address by using the NetScaler GUI:

Navigate to **System** > **Network** > **VXLANs**, and add a new VXLAN entity or modify an existing VXLAN entity.

To add a bridgetable by using the NetScaler GUI:

Navigate to **System** > **Network** > **Bridge Table**, set the following paramters while adding or modifying a VXLAN bridge table entry:

- MAC
- VTEP
- VXLAN ID

To display the VXLAN forwarding table by using the NetScaler GUI:

Navigate to **System** > **Network** > **Bridge Table**.

```
Example
> add vxlan 9000
Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.102.102

Done
```

Support of IPv6 Dynamic Routing Protocols on VXLANs

The NetScaler appliance supports IPv6 dynamic routing protocols for VXLANs. You can configure various IPv6 Dynamic Routing protocols (for example, OSPFv3, RIPng, BGP) on VXLANs from the VTYSH command line. An option IPv6 Dynamic Routing Protocol has been added to VXLAN command set for enabling or disabling IPv6 dynamic routing protocols on a VXLAN. After enabling IPv6 dynamic routing protocols on a VXLAN, processes related to the IPv6 dynamic routing protocols are required to be started on the VXLAN by using the VTYSH command line.

To enable IPv6 Dynamic routing protocols on a VXLAN by using the NetScaler command line:

- **add vxlan** <ID> [-**ipv6DynamicRouting** (**ENABLED** | **DISABLED**)]
- **show vxlan**

```
In the following sample configuration, VXLAN-9000 is created and has
    IPv6 dynamic routing protocols enabled on it. Then, using the VTYSH
    command line, process for the IPv6 OSPF protocol is started on the
    VXLAN.

> add vxlan 9000 -ipv6DynamicRouting ENABLED

Done
> bind vxlan 9000 -srcIP 203.0.100.100

Done
> add bridgetable -mac 00:00:00:00:00:00 -vxlan 9000 -vtep
    203.0.101.101

```

```
Done
> VTYSH
NS# configure terminal
NS(config)# ns IPv6-routing
NS(config)# interface VXLAN-9000
NS(config-if)# ipv6 router OSPF area 3
```

Extending VLANs from Multiple Enterprises to a Cloud using VXLAN-VLAN Maps

CloudBridge Connector tunnels is used to extend an enterprise's VLAN to a cloud. VLANs extended from multiple enterprises can have overlapping VLAN IDs. You can isolate each enterprise's VLANs, by mapping them to a unique VXLAN in the cloud. On a NetScaler appliance, which is the CloudBridge connector endpoint in the cloud, you can configure a VXLAN-VLAN map that links an enterprise's VLANs to a unique VXLAN in the cloud. VXLANs support VLAN tagging for extending multiple VLANs of an enterprise from CloudBridge Connector to the same VXLAN.

Perform the following tasks for extending VLANs of multiple enterprises to a cloud:

1. Create a VXLAN-VLAN map.
2. Bind the VXLAN-VLAN map to a network bridge based or PBR based CloudBridge Connector tunnel configuration on the NetScaler appliance on cloud.
3. (Optional) Enable VLAN tagging in a VXLAN configuration.

NetScaler command line procedures

To add a VXLAN-VLAN map by using the NetScaler command line:

- **add vxlanVlanMap** <name>
- **show vxlanVlanMap** <name>

To bind a VXLAN and VLANS to a VXLAN-VLAN map by using the NetScaler command line:

- **bind vxlanVlanMap** <name> [-**vxlan** <positive_integer> -**vlan** <int[-int]> …]
- **show vxlanVlanMap** <name>

To bind a VXLAN-VLAN map to a network bridge based CloudBridge Connector tunnel by using the NetScaler command line:

At the command prompt, type one of the following sets of commands.

if adding a new network bridge:

- **add netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

if reconfiguring an existing network bridge:

- **set netbridge** <name> [-**vxlanVlanMap** <string>]
- **show netbridge** <name>

To bind a VXLAN-VLAN map to a PBR based CloudBridge Connector tunnel by using the NetScaler command line:

At the command prompt, type one of the following sets of commands.

if adding a new PBR:

- **add pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

if reconfiguring an existing PBR:

- **set pbr** <name> **ALLOW** (-**ipTunnel** <ipTunnelName> [-**vxlanVlanMap** <name>])
- **show pbr** <name>

To include VLAN tags in packets related to a VXLAN by using the NetScaler command line:

At the command prompt, type one of the following sets of commands.

if adding a new VXLAN:

- **add vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

if reconfiguring an existing VXLAN:

- **set vxlan** <vnid> -**vlanTag** (**ENABLED** | **DISABLED**)
- **show vxlan** <vnid>

NetScaler GUI procedures

To add a VXLAN-VLAN map by using the NetScaler GUI:

Navigate to **System** > **Network** > **VXLAN VLAN Map**, add a VXLAN VLAN map.

To bind a VXLAN-VLAN map to a netbridge based CloudBridge Connector tunnel by using the NetScaler GUI:

Navigate to **System** > **CloudBridge Connector** > **Network Bridge**, select a VXLAN-VLAN map from the **VXLAN VLAN** drop down list while adding a new network bridge, or reconfiguring an existing network bridge.

To bind a VXLAN-VLAN map to a PBR based CloudBridge Connector tunnel by using the NetScaler GUI:

Navigate to **System** > **Network** > **PBRs**, on the Policy Based Routing (PBRs) tab, select a **VXLAN-VLAN** map from the **VXLAN VLAN** drop down list while adding a new PBR, or reconfiguring an existing PBR.

To include VLAN tags in packets related to a VXLAN by using the NetScaler GUI:

Navigate to **System** > **Network** > **VXLANs**, enable **Inner VLAN Tagging** while adding a new VXLAN, or reconfiguring an existing VXLAN.

```
> add vxlanVlanMap VXLANVLAN-DC1

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3000 -vlan 3

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 3500 -vlan 4

Done

>add vxlanVlanMap VXLANVLAN-DC2

Done

> bind vxlanvlanmap VXLANVLAN-DC1 -vxlan 8000 -vlan 3 4

Done

> set pbr PBR-CBC-DC-1-CLOUD ALLOW  -ipTunnel  CBC-DC-1-CLOUD -
   vxlanVlanMap VXLANVLAN-DC1

Done

> set pbr PBR-CBC-DC-2-CLOUD ALLOW  -ipTunnel  CBC-DC-2-CLOUD -
   vxlanVlanMap VXLANVLAN-DC2

Done
```

Optimization

September 21, 2020

The NetScaler optimization features reduce transaction times between the clients and the servers, and they reduce bandwidth consumption. They also enhance server performance by offloading some tasks and making others more efficient.

Feature	Description
Client Keep-Alive	Handles multiple requests on a single client connection. The client does not have to negotiate a new connection for each request to the server.
HTTP Compression	Compresses HTTP responses sent from the servers to compression-aware browsers. The smaller responses reduce download time and save bandwidth.
Integrated Caching	Stores responses to client requests. Subsequent requests for the same content are served from the NetScaler cache instead of being forwarded to the origin server.
Front End Optimization	Reduces the load and render time of web pages by simplifying and optimizing the content served to the client browser. **Note**: Supported from NetScaler 10.5 onwards.
Content Accelerator	Stores server responses on a Citrix ByteMobile T2100 appliance. **Note**: Supported from NetScaler 10.1 onwards.
SPDY (Speedy)	Acts as a SPDY gateway between clients and your servers, providing SPDY support without the need to configure/upgrade SPDY on the servers. **Note**: Supported from NetScaler 10.1 onwards.

Client Keep-Alive

September 21, 2020

The client keep-alive feature enables multiple client requests to be sent on a single client connection. This feature helps in a transaction management environment where typically the server closes the client connection after serving the response. The client then opens a new connection for each request and spends more time on the transaction.

Client keep-alive resolves this issue by keeping the connection between the client and the appliance (client-side connection) open even after the server closes the connection with the appliance. This allows sending multiple client requests using a single connection and saves the round trips associated with opening and closing a connection. Client keep-alive is most beneficial in SSL sessions.

Client keep-alive is also useful under either of the following conditions:

- When the server does not support client keep-alive.
- When the server supports client keep-alive but an application on the server does not support client keep-alive.

> **Note**
> Client keep-alive is applicable for HTTP and SSL traffic. Client-keep alive can be configured globally to be able to handle all traffic. It can also be configured to be active only on specific services.

In client keep-alive environment, the configured services intercept the client traffic and the client request is directed to the origin server. The server sends the response and closes the connection between the server and the appliance. If a "Connection: Close" header is present in the server response, the appliance corrupts this header in the client-side response, and the client-side connection is kept open. As a result, the client does not have to open a new connection for the next request; instead, the connection to the server is reopened.

> **Note**
> If a server sends back two "Connection: Close" headers, only one is edited. This results in significant delays on the client rendering of the object because a client does not assume that the object has been delivered completely until the connection is actually closed.

Configure client keep-alive

Client keep-alive, by default, is disabled on the NetScaler, both globally and at service level. Therefore, you must enable the feature at the required scope.

> **Note**
> If you enable client keep-alive globally, it is enabled for all services, regardless of whether you enable it at the service level. Additionally, if required, you can configure some HTTP parameters to specify the maximum number of HTTP connections retained in the connection reuse pool, enable connection multiplexing, and enable persistence Etag.
> **Note**
> When Persistent ETag is enabled, the ETag header includes information about the server that served the content. This ensures that cache validation conditional requests or browser requests, for that content, always reaches the same server.

To configure client keep-alive by using the command line interface:

At the command prompt, do the following:

1. Enable client keep-alive on the NetScaler.

- At global level

  `enable ns mode cka`

- At service level

  `set service <name> -CKA YES`

> **Note**:
> Client keep-alive can be enabled only for HTTP and SSL services.

2. Configure the required HTTP parameters on the HTTP profile that is bound to the service(s).

```
```
set ns httpProfile <name> -maxReusePool <value> -conMultiplex ENABLED -
    persistentETag ENABLED
<!--NeedCopy--> ```

>**Note**: Configure these parameters on the `
    nshttp_default_profile HTTP` profile, to make them available
    globally.
```

To configure client keep-alive by using the GUI:

1. Enable client keep-alive on the NetScaler.
   - At global level

     Navigate to **System** > **Settings**, click **Configure Modes** and select **Client side Keep Alive**.

   - At service level

     Navigate to **Traffic Management** > **Load Balancing** > **Services**, and select the required service. In the **Settings** grouping, enable **Client Keep-Alive**.

2. Configure the required HTTP parameters on the HTTP profile that is bound to the service(s).

   Navigate to **System** > **Profiles**, and on **HTTP Profiles** tab, select the required profile and update the required HTTP parameters.

HTTP compression

September 21, 2020

For websites with compressible content, the HTTP compression feature implements lossless compression to alleviate latency, long download times, and other network-performance problems by compressing the HTTP responses sent from servers to compression-aware browsers. You can improve
```

server performance by offloading the computationally intensive compression task from your servers to the NetScaler appliance.

The following table describes the capabilities of the HTTP compression feature:

| Functionality | Description |
|---|---|
| Compression Ratio | Compression ratio depends on the types of files in the responses, but is always significant, noticeably reducing amount of data transmitted over the network. |
| Browser Awareness | NetScaler serves compressed data to compression aware browsers only, reducing the transaction time between the client and the server. Most modern web browsers support HTTP compression. |
| Compression blocking | You can define content filters to selectively block compression by applying built-in actions. |
| Compression Caching | With the integrated caching feature enabled, subsequent requests for the same content are served from the local cache, reducing the number of round trips to the server and improving transaction times. |
| HTTPS Support | Compression is particularly useful on SSL connections, because it reduces the amount of content that has to be encrypted, either on the server or by the NetScaler appliance, and decrypted by the client. |
| Intelligent Response Filtering | The NetScaler compression engine intelligently filters server responses on the basis of defined compression parameters. For example, the compression engine detects zero-content-length responses and compressed responses and does not compress them. The detection of compressed responses enables origin sites to use server-based compression in conjunction with the NetScaler compression feature. |

| Functionality | Description |
|---|---|
| Compression Switching | The NetScaler appliance transparently directs requests from compression aware clients to compression capable servers, so that responses to those clients are compressed, and responses to other clients are not delayed by compression processing. |

### How HTTP compression works

A NetScaler can compress both static and dynamically generated data. It applies the GZIP or the DEFLATE compression algorithm to remove extraneous and repetitive information from the server responses and represent the original information in a more compact and efficient format. This compressed data is sent to the client's browser and uncompressed as determined by the browser's supported algorithm or algorithms (GZIP or DEFLATE).

NetScaler compression treats static and dynamic content differently.

- Static files are compressed only once, and a compressed copy is stored in local memory. Subsequent client requests for cached files are serviced from that memory.
- Dynamic pages are dynamically created each time a client requests them.

When a client sends a request to the server:

1. The client request arrives at the NetScaler. The appliance examines the headers and stores information about what kind of compression, if any, the browser supports.
2. The appliance forwards the request to the server and receives the response.
3. The NetScaler compression engine examines the server response for compressibility by matching it against policies.
4. If the response matches a policy associated with a compression action, and the client browser supports a compression algorithm specified by the action, the NetScaler applies the algorithm and sends the compressed response to the client browser.
5. The client applies the supported compression algorithm to decompress the response.

### Configure HTTP compression

By default, compression is disabled on the NetScaler. You must enable the feature before configuring it. If the feature is enabled, the appliance compresses server requests specified by compression policies.

To enable HTTP compression by using the CLI

Compression can be enabled for HTTP and SSL services only. You can enable it globally, so that it applies to all HTTP and SSL services, or you can enable it just for specific services.

At the command prompt, enter one of the following commands to enable compression globally or for a specific service:

- `enable ns feature cmp`
  OR
- `set service <name> -CMP YES`

To configure compression by using the GUI

Do one of the following:

To enable compression globally, navigate to System > Settings, click Configure Basic Features, and select HTTP Compression.

To enable compression for a specific service, navigate to Traffic Management > Load Balancing > Services, select the service, and click Edit. In the Settings group, click the pencil icon and enable Compression.

### Configuring a compression action

A compression action specifies the action to take when a request or response matches the rule (expression) in the policy with which the action is associated. For example, you can configure a compression policy that identifies requests that will be sent to a particular server, and associate the policy with an action that compresses the server's response.

There are four built-in compression actions:

- COMPRESS: Uses the GZIP algorithm to compress data from browsers that support either GZIP or both GZIP and DEFLATE. Uses the DEFLATE algorithm to compress data from browsers that support only the DEFLATE algorithm. If the browser does not support either algorithm, the browser's response is not compressed.
- NOCOMPRESS: Does not compress data.
- GZIP: Uses the GZIP algorithm to compress data for browsers that support GZIP compression. If the browser does not support the GZIP algorithm, the browser's response is not compressed.
- DEFLATE: Uses the DEFLATE algorithm to compress data for browsers that support the DEFLATE algorithm. If the browser does not support the DEFLATE algorithm, the browser's response is not compressed. After creating an action, you associate the action with one or more compression policies.

At the command prompt, enter the following command to create a compression action:

```
add cmp action <name> <cmpType> [-addVaryHeader <addVaryHeader> -varyHeaderValue
<string>]
```

To configure a compression policy by using the CLI

A compression policy contains a rule, which is a logical expression that enables the NetScaler appliance to identify the traffic that should be compressed.

When the NetScaler receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.

At the command prompt, enter the following command to create a compression policy:

```
add cmp policy <name> -rule <expression> -resAction <string>
```

To create a compression action by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Actions** , click **Add** , and create a compression action to specify the type of compression to be performed on the HTTP response.

Configuring a compression policy

A compression policy contains a rule, which is a logical expression that enables the NetScaler appliance to identify the traffic that should be compressed.

When the NetScaler receives an HTTP response from a server, it evaluates the built-in compression policies and any custom compression policies to determine whether to compress the response and, if so, the type of compression to apply. Priorities assigned to the policies determine the order in which the policies are matched against the requests.

The following table lists the built-in HTTP compression policies. These policies are activated globally when you enable compression.

Built-in Classic or Default Syntax Policy	Description
ns_nocmp_mozilla_47, ns_adv_nocmp_mozilla_	Prevents compression of CSS files when a request is sent from a Mozilla 4.7 browser.
ns_cmp_mscss, ns_adv_cmp_mscss	Compresses CSS files when the request is sent from a Microsoft Internet Explorer browser.
ns_cmp_msapp, ns_adv_cmp_msapp	Compresses files that are generated by the following applications: Microsoft Office Word, Microsoft Office Excel, Microsoft Office PowerPoint.
ns_cmp_content_type, ns_adv_cmp_content_type	Compresses data when the response containsContent-Type header and contains text.

Built-in Classic or Default Syntax Policy	Description
ns_nocmp_xml_ie, ns_adv_nocmp_xml_ie	Prevents compression when a request is sent, from a Microsoft Internet Explorer browser and the response contains a Content-Type header and contains text or xml.

Binding a compression policy

To put a compression policy into effect, you must bind it either globally, so that it applies to all traffic that flows through the NetScaler, or to a specific virtual server, so that the policy applies only to requests whose destination is the VIP address of that virtual server.

When you bind a policy, you assign it a priority. The priority determines the order in which the policies you define are evaluated. You can set the priority to any positive integer.

To bind a Compression Policy by using the CLI

At the command prompt, enter one of the following commands to bind a compression policy globally or to a specific virtual server:

- `bind cmp global <policyName> [-priority <positive_integer>] [-state ( ENABLED|DISABLED)]...`
- `bind lb vserver <vserverName> -policyName <policyName> -priority < positive_integer>.`

Repeat this command for each virtual server to which you want to bind the compression policy.

To bind a compression policy by using the GUI

Do one of the following:

At global level Navigate to **Optimization** > **HTTP Compression** > **Policies**, click **Policy Manager** and bind the required policies by specifying the relevant Bind Point and Connection Type (Request/Response).

At virtual server level

For load balancing virtual server, Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select the required virtual server, click **Policies**, and bind the relevant policy.

For content switching virtual server, Navigate to Traffic Management > Content Switching > Virtual Servers, select the required virtual server, click Policies, and bind the relevant policy.

Set the Global Compression Parameters for Optimal Performance

Many users accept the default values for the global compression parameters, but you might be able to provide more effective compression by customizing these settings.

> **Note**
> After you configure the global compression parameters, you do not have to reboot your appliance. They get applied to the new flows immediately.

The following table describes the compression parameters that you can set on the NetScaler.

Compression Parameters	Description
Quantum size	Size, in KB, of the buffer maintained for accumulating server responses. The responses are compressed when the buffer size exceeds this value. For example, if you set the quantum size to 50 KB, the NetScaler compresses the buffer's contents when its size becomes larger than 50 KB. Minimum value: 1. Maximum value: 63488. Default: 57344.
Compression level	Level of compression to apply to server responses. Possible values: Best Speed, Best Compression, optimal.
Minimum HTTP response size	Minimum size, in bytes, of an HTTP response that is compressed. Responses smaller than the value specified by this parameter are sent without being compressed.
Bypass compression on CPU usage	NetScaler CPU usage, as a percentage, at or above which no compression is done. Default: 100.
Policy Type*	Type of policies used for compression. Possible values: Classic, Default Syntax. Default: Classic.
Allow Server-side compression	Allow servers to send compressed data to the NetScaler.
Compress push packet	Upon receipt of a packet with a TCP PUSH flag, compress the accumulated packets immediately, without waiting for the quantum buffer to be filled.

Compression Parameters	Description
External Cache	Issue a private response directive indicating that the response message is intended for a single user and must not be cached by a shared or proxy cache.

To configure HTTP compression by using the GUI

Do one of the following:

- To enable compression globally, navigate to **System** > **Settings**, click **Configure Basic Features**, and select **HTTP Compression**.
- To enable compression for a specific service, navigate to **Traffic Management** > **Load Balancing** > **Services**, select the service, and click **Edit**.
- In the **Settings** group, click the pencil icon and enable **Compression**.

To create a compression action by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Actions** , click **Add** , and create a compression action to specify the type of compression to be performed on the HTTP response

To create a compression policy by using the GUI

Navigate to **Optimization** > **HTTP Compression** > **Policies** , click **Add**, and create a compression policy by specifying the condition and the corresponding action to be executed.

Evaluate compression configuration

You can view the compression statistics in the dashboard utility or in an SNMP monitor. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

Optionally, you can also view statistics for a compression policy, including the number of hits that the policy counter increments during the policy based compression.

> **Note**
>
> - For more information about the statistics and charts, see the Dashboard help on the NetScaler appliance.
> - For more information about SNMP, see SNMP topic.

To view compression statistics by using the CLI

At the command prompt, enter the following commands to display the compression statistics:

1. To display compression statistics summary.

```
stat cmp
```

> **Note**
> The stat cmp policy command displays statistics for default syntax compression policies only.

2. To display compression policy hits and details

```
show cmp policy <name>
```

3. To display detailed compression statistics

```
stat cmp -detail
```

To view compression statistics by using the dashboard:

In the Dashboard utility, you can display the following types of compression statistics:

- Select Compression to display a summary of the compression statistics.
- To display detailed compression statistics by protocol type, click the Details
- To display the rate of requests processed by the compression feature, click the Graphical View tab.

To view compression statistics by using SNMP

You can view the following compression statistics by using the SNMP network management application.

- Number of compression requests (OID: 1.3.6.1.4.1.5951.4.1.1.50.1)
- Number of compressed bytes transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.2)
- Number of compressible bytes received (OID: 1.3.6.1.4.1.5951.4.1.1.50.3)
- Number of compressible packets transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.4)
- Number of compressible packets received (OID: 1.3.6.1.4.1.5951.4.1.1.50.5)
- Ratio of compressible data received and compressed data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.6)
- Ratio of total data received to total data transmitted (OID: 1.3.6.1.4.1.5951.4.1.1.50.7)

To view additional compression statistics by using the GUI

1. To display HTTP compression statistics:

Navigate to **Optimization** > **HTTP Compression** and click **Statistics**.

2. To display statistics of a compression policy.

Navigate to **Optimization** > **HTTP Compression** > **Policies** > select the policy, and click **Statistics**.

3. To display statistics of a compression policy label

Navigate to **Optimization** > **HTTP Compression** > **Policies** > select a policy label, and click **Statistics**.

Offloading HTTP compression

Performing compression on a server can affect the server's performance. A NetScaler placed in front of your web servers and configured for HTTP compression offloads compression of both static and dynamic content, saving server CPU cycles and resources.

You can offload compression from the Web servers in either of two ways:

Disable compression on the web servers, enable the NetScaler Compression feature at a global level, and configure services for compression.

Leave the compression feature enabled on the web servers and configure the NetScaler appliance to remove the "Accept Encoding" header from all HTTP client requests. The servers then send uncompressed responses. The NetScaler compresses the server responses before sending them to the clients.

> **Note**
> The second option does not work if the servers automatically compress all responses. The NetScaler does not attempt to compress a response that is already compressed.

The Servercmp parameter enables the NetScaler appliance to handle offload HTTP compression. By default, this parameter is set ON for the server to send compressed data to the NetScaler appliance. To offload HTTP compression, you need to set the servercmp parameter to OFF. At the command prompt, enter the following commands:

```
set service <service name> -CMP YES
```

Repeat this command for each service for which you want to enable compression.

```
show service <service name>
```

Repeat this command for each service, to verify that compression is enabled.

```
Save config
```

```
set cmp parameter -serverCmp OFF
```

> **Note**:
>
> When `Servercmp` parameter is turned on and if the appliance receives compressed response from the server, the appliance does not further compress the data. Instead, it forwards the compressed response to the client.

Integrated caching

September 21, 2020

The integrated cache provides in-memory storage on the Citrix ADC appliance and serves Web content to users without requiring a round trip to an origin server. For static content, the integrated cache requires little initial setup. After you enable the integrated cache feature and perform basic setup (for example, determining the amount of Citrix ADC appliance memory the cache is permitted to use), the integrated cache uses built-in policies to store and serve specific types of static content, including simple webpages and image files. You can also configure the integrated cache to store and serve dynamic content that is marked as non-cacheable by Web and application servers (for example, database records and stock quotes).

Note: The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

When a request or response matches the rule (logical expression) specified in a built-in policy or a policy that you have created, the Citrix ADC appliance performs the action associated with the policy. By default, all policies store cached objects in and retrieve them from the Default content group, but you can create your own content groups for different types of content.

To enable the appliance to find cached objects in a content group, you can configure selectors. The selectors match cached objects against expressions, or you can specify parameters for finding objects in the content group. If you use selectors as recommended by Citrix, configure them first, so that you can specify selectors when you configure content groups. Next, set up any content groups that you want to add, so that they are available when you configure the policies. To complete the initial configuration, create policy banks by binding each policy to a global bind point or a virtual server. Or, you can bind a label that can be called from other policy banks.

Integrated caching can be improved using pre-loading cached object method before they are scheduled to expire. To manage the handling of cached data, you can configure caching-related headers inserted into the responses. The integrated cache can also act as a forward proxy for other cache servers.

> **Note:**
>
> Integrated caching requires some familiarity with HTTP requests and responses. For information about the structure of HTTP data, see *Live HTTP Headers* at `"<http://livehttpheaders.mozdev.org/>."`

How integration cache works

The integrated cache monitors HTTP and SQL requests that flow through the Citrix ADC appliance and compares the requests with stored policies. Depending on the outcome, the integrated cache feature either searches the cache for the response or forwards the request to the origin server. For HTTP requests, integrated caching serves as partial content from the cache in response to single byte-range and multi-part byte-range requests.

Cached data is compressed if the client accepts compressed content. You can configure expiration times for a content group, and you can selectively expire entries in a content group.

Data that is served from the integrated cache is a `hit`, and data served from the origin is a cache miss, as described in the following table.

Transaction Type	Specification
Cache Hit	Responses that the Citrix ADC appliance serves from the cache, including: Static objects, for example, image files and static webpages, 200 OK pages, 203 Non-Authoritative Response pages, 300 Multiple Choices pages, 301 Moved Permanently pages, 302 Found pages, 304 Not Modified pages, These responses are known as positive responses. The Citrix ADC appliance also caches the following negative responses: 307 Temporary Redirect pages, 403 Forbidden pages, 404 Not Found pages, 410 Gone pages. To further improve performance, you can configure the Citrix ADC appliance to cache more types of content.
Storable Cache Miss	For a storable cache miss, the Citrix ADC appliance fetches the response from the origin server, and stores the response in the cache before serving it to the client.
Non-Storable Cache Miss	A non-storable cache miss is inappropriate for caching. By default, any response that contains the following status codes is a non-storable cache miss: 201, 202, 204, 205, 206 status codes, All 4xx codes, except 403, 404 and 410, 5xx status codes

> **Note:**
> To integrate dynamic caching with your application infrastructure, use the NITRO API to issue cache commands remotely. For example, you can configure triggers that expire cached responses when a database table is updated.

To ensure the synchronization of cached responses with the data on the origin server, you configure expiration methods. When the Citrix ADC appliance receives a request that matches an expired response, it refreshes the response from the origin server.

> **Note:**
> Citrix recommends that you synchronize the times on the Citrix ADC appliance and one or more back-end servers.

How dynamic cache works

Dynamic caching evaluates HTTP requests and responses based on parameter-value pairs, strings, string patterns, or other data. For example, suppose that a user searches for Bug 31231 in a bug reporting application. The browser sends the following request on the user's behalf:

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000

Host: mycompany.net

User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.9) Gecko/2008052906 Firefox/3.0

Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8

Accept-Language: en-us,en;q=0.5

. . .
<!--NeedCopy-->
```

In this example, GET requests for this bug reporting application always contain the following parameters:

- IssuePage
- RecordID
- Template
- TableId

GET requests do not update or alter the data, so you can configure these parameters in caching policies and selectors, as follows:

- You configure a caching policy that looks for the string mybugreportingsystem and the GET method in HTTP requests. This policy directs matching requests to a content group for bugs.
- In the content group for bugs, you configure a `hit` selector that matches various parameter-value pairs, including IssuePage, RecordID, and so on.

> **Note:** that a browser can send multiple GET requests based on one user action. The following is a series of three separate GET requests that a browser issues when a user searches for a bug based on a bug ID.

```
GET /mybugreportingsystem/mybugreport.dll?IssuePage&RecordId=31231&Template=view&TableId=1000

GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbtns&RecordId=31231&TableId=1000

GET /mybugreportingsystem/mybugreport.dll?IssuePage&Template=viewbody&
    RecordId=31231&tableid=1000
<!--NeedCopy-->
```

To fulfill these requests, multiple responses are sent to the user's browser, and the webpage that the user sees is an assembly of the responses.

If a user updates a bug report, the corresponding responses in the cache should be refreshed with data from the origin server. The bug reporting application issues HTTP POST requests when a user updates a bug report. In this example, you configure the following to ensure that POST requests trigger invalidation in the cache:

- A request-time invalidation policy that looks for the string mybugreportingsystem and the POST HTTP request method, and directs matching requests to the content group for bug reports.
- An invalidation selector for the content group for bug reports that expires cached content based on the RecordID parameter. This parameter appears in all of the responses, so the invalidation selector can expire all relevant items in the cache.

The following excerpt shows a POST request that updates the sample bug report.

```
POST /mybugreportingsystem/mybugreport.dll?TransitionForm HTTP/1.1\r\n

User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0) Opera
    7.23 [en]\r\n

Host: mybugreportingsystem\r\n

Cookie:ttSearch.134=%23options%3Afalse%23active%23owner%3Afalse%23
    unowned%3Afalse%23submitter%3Afalse%23incsub%3Atrue;

Cookie2: $Version=1\r\n

. . .

\r\n

ProjectId=2&RecordId=31231&TableId=1000&TransitionId=1&Action=Update&
    CopyProjectId=0&ReloadForm=0&State=&RecordLockId=49873+issues+in+
    HTTP&F43. . .
<!--NeedCopy-->
```

When the Citrix ADC appliance receives this request, it does the following:

- Matches the request with an invalidation policy.

- Finds the content group that is named in the policy.
- Applies the invalidation selector for this content group and expires all responses that match RecordID=31231.

When a user issues a new request for this bug report, the Citrix ADC appliance goes to the origin server for updated copies of all the responses that are associated with the report instance, stores the responses in the content group, and serves them to the user's browser, which reassembles the report and displays it.

Configure integrated cache

To use the integrated cache, you must install the license and enable the feature. After you enable the integrated cache, the Citrix® Citrix ADC® appliance automatically caches static objects as specified by built-in policies and generates statistics on cache behavior. (Built-in policies have an underscore in the initial position of the policy name.)

Even if the built-in policies are adequate for your situation, you might want to modify the global attributes. For example, you might want to modify the amount of Citrix ADC appliance memory allocated to the integrated cache.

If you would like to observe cache operation before changing settings, see "Displaying Cached Objects and Cache Statistics."

> **Note**
> The Citrix ADC cache is an in-memory store that is purged when you restart the appliance.

To install integrated cache license

An integrated cache license is required. For information about licenses, see information about obtaining licenses topic.

Obtain a license code from Citrix, go to the command line interface, and log in.

At the command line interface, copy the license file to the `/nsconfig/license` folder.

Reboot the Citrix ADC appliance by using the following command:

```
reboot
```

To enable integrated caching
When you enable integrated caching, the Citrix ADC appliance begins caching server responses. If you have not configured any policies or content groups, the built-in policies store cached objects in the Default content group.

At the command prompt, type one of the following commands to enable or disable integrated caching:

'enable ns feature IC'

To configure global attributes for caching

Global attributes apply to all cached data. You can specify the amount of Citrix ADC memory allocated to the integrated cache, Via header insertion, a criterion for verifying that a cached object should be served, the maximum length of a POST body permitted in the cache, whether to bypass policy evaluation for HTTP GET requests, and an action to take when a policy cannot be evaluated.

The cache memory capacity is limited only by the memory of the hardware appliance. Also, any packet engine (the central distribution hub of all incoming TCP requests) in the nCore Citrix ADC appliance is aware of objects cached by other packet engines in the nCore Citrix ADC appliance.

> **Note:**
> When the default global memory limit is set as 0 and the Integrated Caching (IC) feature is enabled, the appliance does not cache any objects. For caching, you must explicitly configure the global memory limit. However, if you enable "set authentication, authorization, and auditing parameter enableStaticPageCaching" option, there will be some default memory configured in the appliance. This memory is insufficient for caching large Objects and so it is necessary to assign a higher memory limit for IC. You can perform this by configuring the "set cache parameter –memLimit" command. The new setting is applied only after you save the configuration and reboot the appliance.

You can modify the global memory limit configured for caching objects. However, when you update the global memory limit to a value lower than the existing value (for example, from 10 GB to 4 GB), the appliance continues to use the memory limit.

This means that even though the integrated caching limit is configured to some value, the actual limit used can be higher. This excessive memory is however released when the objects are removed from cache.

The output of the show cache parameter command indicates the configured value (memory Usage limit) and the actual value being used (memory usage limit (active value)).

At the command prompt, type:

```
set cache parameter [-memLimit <MBytes>] [-via <string>] [-verifyUsing <
criterion>] [-maxPostLen <positiveInteger>] [-prefetchMaxPending <positiveInteger
>] [-enableBypass(YES|NO)] [-undefAction (NOCACHE|RESET)]
```

To enable integrated caching by using the GUI:

Navigate to **System** > **Settings**, click **Configure Basic Features**, and select **Integrated Caching**.

To configure global settings for caching by using the GUI:

Navigate to **Optimization** > **Integrated Caching**, click **Change Cache Settings**, and configure the global settings for caching.

To set up built-in content group, pattern set, and policies for Integrated Cache:

The Citrix ADC appliance includes a built-in integrated caching configuration that you can use for caching content. The configuration consists of a content group called ctx_cg_poc, a pattern set called ctx_file_extensions, and a set of integrated cache policies. In the content group ctx_cg_poc, only objects that are 500 KB or smaller are cached. The content is cached for 86000 seconds, and the memory limit for the content group is 512 MB. The pattern set is an indexed array of common extensions for file-type matching.

The following table lists the built-in integrated caching policies. By default, the policies are not bound to any bind point. You must bind the policies to a bind point if you want the Citrix ADC appliance to evaluate traffic against the policies. The policies cache objects in the ctx_cg_poc content group.

Integrated caching policy name	Policy rule
_cacheVPNStaticObjects	HTTP.REQ.URL.SET_TEXT_MODE(IGNORECASE).CONTAINS_IN
_cacheTCPVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(".css")
_cacheOCVPNStaticObjects	HTTP.REQ.URL.ENDSWITH(".pdf")
_cacheWFStaticObjects	HTTP.REQ.URL.ENDSWITH(".js")
_mayNoCacheReq	HTTP.RES.HEADER("Content-Type").CONTAINS("application/x-javascript")
_noCacheRest	TRUE

Flush cache configuration

You can flush a cache group, cache groups, or cache object locator. Following are the commands to flush cache objects.

At the command prompt, type:

`flush cache contentgroup all`

Example

```
0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.html?name=hello
0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.html?name=hi

Flush cache contentGroup all
done
<!--NeedCopy-->
```

```
flush cache contentgroup <content group name>
```

Example

```
0x00000089bae000000004 DEFAULT GET  //1.1.1.1:80/html/index.html?name=hello
0x00000089bae000000005 DEFAULT GET  //1.1.1.1:80/html/index.html?name=hi

Flush cache ob -| 0x00000089bae000000004
done
<!--NeedCopy-->
```

```
flush cache object (-locator <positive_integer> | (-url <URL> (-host <
string> [-port <port>] [-groupName <string>] [-httpMethod ( GET | POST )
])))
```

Example

```
0x00000089bae000000006 DEFAULT GET //1.1.1.1:80/html/index.html

flush cache ob -URL /html/index.html -host 1.1.1.1 -groupName DEFAULT
done
<!--NeedCopy-->
```

Front end optimization

September 21, 2020

> **Note:** Front end optimization is available if you have an Advanced or Premium Citrix ADC license and are running Citrix ADC release 10.5 or later.

The HTTP protocols that underlie web applications were originally developed to support transmission and rendering of simple web pages. New technologies such as JavaScript and cascading style sheets (CSS), and new media types such as Flash videos and graphics-rich images, place heavy demands on front-end performance, that is, on performance at the browser level.

The Citrix ADC front end optimization (FEO) feature addresses such issues and reduces the load time and render time of web pages by:

- Reducing the number of requests.

- Required for rendering each page.
- Reducing the number of bytes in page responses.

Simplifying and optimizing the content served to the client browser.

You can customize your FEO configuration to provide the best results for your users. Citrix ADCs support numerous web content optimizations for both desktop and mobile users. The following tables describe the front-end optimizations provided by the FEO feature, and the operations performed on different types of files.

Optimizations performed by the FEO feature

Web Optimization	Problem	What Citrix ADC FEO feature does	Benefits
Inlining	Client browsers often send multiple requests to servers for loading external CSS, images, and JavaScript associated with the web page.	CSS inline, JavaScript inline, CSS combine	Loading the external CSS, images, and JavaScript inline with the HTML files improves page-rendering time. This optimization is beneficial for content that will be viewed only once, and for mobile devices that have limited cache sizes.
Minification	Data fetched from servers includes inessential characters such as white spaces, comments, and newline characters. The time that browsers spend in processing such data creates website latency.	CSS minification, JavaScript minification, Removal of HTML comments	Minified files consume less bandwidth and avoid the latency caused by special processing.

Web Optimization	Problem	What Citrix ADC FEO feature does	Benefits
Image optimization	Mobile browsers often have slow connection speeds and limited cache memory. Downloading images on mobile clients consumes more bandwidth, processing time, and cache space, resulting in web site latency.	JPEG optimization, CSS image inlining, Image shrink-to attributes, GIF to PNG conversion, HTML image inlining, WebP image conversion, JPEG, GIF, PNG to JPEG-XR image conversion	Reduces the image to the size indicated in image tag by Citrix ADC, enabling client browsers to load images faster.
Repositioning	Inefficient processing of external CSS, images, and JavaScript increases page-load time.	Image lazy loading, CSS move to Head, JavaScript move to end	Repositions HTML elements, to reduce rendering time for web pages and enable client browsers to load the objects faster.
Connection Management	Many browsers set limits on the number of simultaneous connections that can be established to a single domain. This can cause browsers to download webpage resources one at a time, resulting in higher browsers time.	Domain sharding	Overcomes the connection limitation, which improves page-rendering time by enabling client browsers to download more resources in parallel.

Web Optimizations performed on different file types:

Citrix ADC can perform web optimizations on CSS, images, Javascript, and HTML. For more information, see Web Optimizations pdf.

> **Note**:
> The front end optimization feature supports ASCII characters only. It does not support the unicode character set.

How front end optimization works

After the Citrix ADC receives the response from the server:

1. Parses the contents of the page, creates an entry in the cache (wherever applicable), and applies the FEO policy.

   For example, a Citrix ADC can apply the following optimization rules:

   - Remove white spaces or comments present within a CSS or JavaScript.
   - Combine one or more CSS files to one file.
   - Convert GIF image format to PNG format.

2. Rewrites the embedded objects and saves the optimized content in the cache, with a different signature than the one used for the initial cache entry.

3. For subsequent requests, fetches the optimized objects from the cache, not from the server, and forwards the responses to the client.

Remove extraneous information such as white spaces and comments.

**

Remove extraneous information such as white spaces and comments.

The period during which the browser can use the cached resource without checking to see if fresh content is available on the server.

Configure front end optimization

Optionally, you can change the values of the front end optimization global settings. Otherwise, begin by creating actions that specify the optimization rules to be applied to the embedded objects.

After configuring actions, create policies, each with a rule specifying a type of request for which to optimize the response, and associate the actions with the policies.

> **Note**: The Citrix ADC evaluates front end optimization policies at request time only, not at response time.

To put the policies into effect, bind them to bind points. You can bind a policy globally, so that it applies to all traffic that flows through the Citrix ADC, or you can bind the policy to a load balancing or content switching virtual server of type HTTP or SSL. When you bind a policy, assign it a priority. A lower priority number indicates a higher value. The Citrix ADC applies the policies in the order of their priorities.

Prerequisites

Front end optimization requires the Citrix ADC integrated caching feature to be enabled. Additionally, you must perform the following integrated caching configurations:

- Allocate cache memory.
- Set the maximum response size and memory limit for a default cache content group.

For more information on configuring integrated caching, see Integrated Caching.

Note: The term Integrated Cache can be interchangeably used with AppCache; note that from a functionality point of view, both terms mean the same.

To configure front end optimization by using the command line interface

At the command prompt, do the following:

1. Enable the front end optimization feature.

`enable ns feature FEO`

1. Create one or more front end optimization actions.

`add feo action <name> [-imgShrinkToAttrib] [-imgGifToPng] ...`

Example: To add a front end optimization action for converting images in GIF format to PNG format and to extend cache expiry period:

`add feo action allact -imgGifToPng -pageExtendCache`

1. [Optional] Specify non-default values for front end optimization global settings.

`set feo parameter [-cacheMaxage <integer>] [-JpegQualityPercent <integer>] [-cssInlineThresSize <integer>] [-inlineJsThresSize <integer> [-inlineImgThresSize <integer>]`

Example: To specify the cache maximum expiry period:

`set feo parameter -cacheMaxage 10`

1. Create one or more front end optimization policy.

`add feo policy <name> <rule> <action>`

Example: To add a front end optimization policy and associate it with the above specifiedallact action:

‘>add feo policy pol1 TRUE allact

> add feo policy pol1 “(HTTP.REQ.URL.CONTAINS(”testsite”))” allact1‘

5. Bind the policy to a load balancing or content switching virtual server, or bind it globally.

```
bind lb vserver <name> -policyName <string> -priority <num>

bind cs vserver <name> -policyName <string> -priority <num>

bind feo global <policyName> <priority> -type <type> <
    gotoPriorityExpression>
<!--NeedCopy-->
```

Example: To apply the front end optimization policy to a virtual server named "abc":

`> bind lb vserver abc -policyName pol1 -priority 1 -type NONE`

Example: To apply the front end optimization policy for all the traffic reaching the ADC:

`> bind feo global pol1 100 -type REQ_DEFAULT`

1. Save the configuration. save ns config

Configure front end optimization by using the GUI

1. Navigate to **Optimization** > **Front End Optimization** > **Actions**, click **Add** and create a front end optimization action by specifying the relevant details.
2. [Optional] Specify the front end optimization global settings.
3. Navigate to **Optimization** > **Front End Optimization**, and on the right-pane, under Settings, click **Change Front End Optimization** settings and specify the front end optimization global settings.
4. Create a front end optimization policy.
5. Navigate to **Optimization** > **Front End Optimization** > **Policies**, click **Add** and create a front end optimization policy by specifying the relevant details.
6. Bind the policy to a load balancing or content switching virtual server.
   a) Navigate to **Optimization** > **Front End ptimization** > **Policies**.
   b) Select a front end optimization policy and click **Policy Manager**.
   c) Under **Front End Optimzation Policy Manager**, bind the front end optimization policy to a load balancing or content switching virtual server.

Verify front end optimization configuration

The dashboard utility displays summary and detailed statistics in tabular and graphic formats. You can view the FEO statistics to evaluate your FEO configuration.

Optionally, you can also display statistics for an FEO policy, including the number of hits that the policy counter increments during policy based FEO.

> **Note**
>
> For more information about statistics and charts, see the Dashboard help on the Citrix ADC appliance.

View FEO statistics by using the command line interface

At the command prompt, type the following commands to display a summary of FEO statistics, FEO policy hits and details, and detailed FEO statistics, respectively:

- `stat feo` Note: The stat feo policy command displays statistics only for advanced FEO policies.
- `show feo policy name`
- `stat feo -detail`

View FEO statistics on Citrix ADC dashboard

In the dashboard GUI, you can:

- Select Front End Optimization to display a summary of FEO statistics.
- Click the **Graphical View** tab to display the rate of requests processed by the FEO feature.

Sample optimization:

Refer to the Sample pdf for some examples of content optimization actions that are applied on HTML content and the embedded objects within the HTML content.

Content accelerator

September 21, 2020

> **Important**
>
> The content accelerator feature is no longer supported on the NetScaler appliance.

Content accelerator is a NetScaler feature that you can use in a Citrix ByteMobile T1100 deployment, to store data on a Citrix ByteMobile T2100 appliance. For more information about Citrix ByteMobile documentation.

Storing data on a T2100 appliance saves bandwidth and provides faster response times, because the NetScaler does not have to connect to the server for repeated requests of the same data.

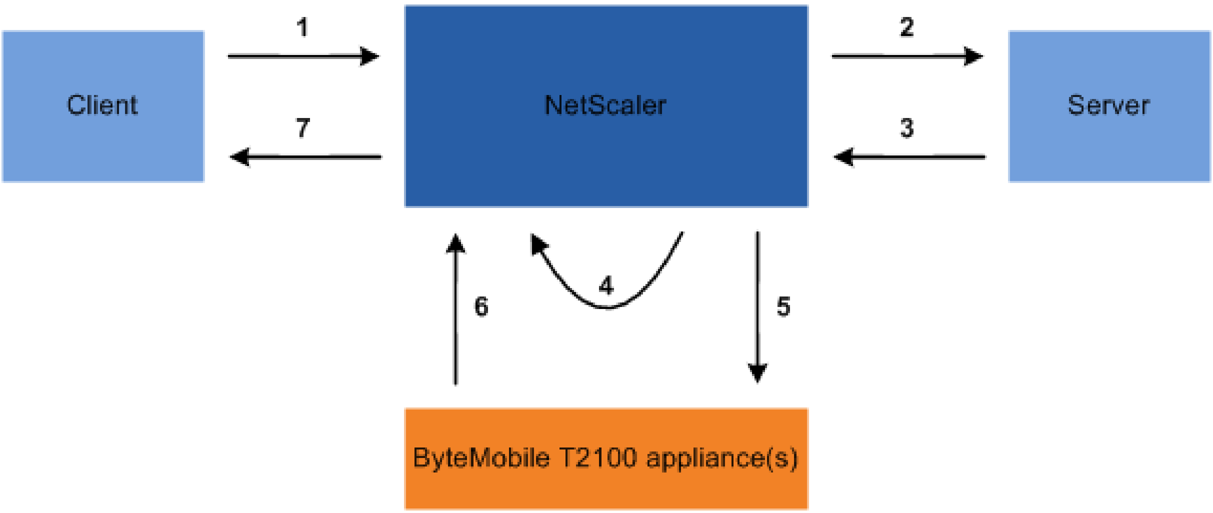
Note: Content accelerator works with a Citrix ByteMobile platinum license. Contact customer support for more information and for obtaining the license.

How content accelerator works

When a load balancing or content switching virtual server receives a client request, the NetScaler appliance evaluates a content accelerator policy that you have bound to the virtual server. The policy filters the requests to identify the ones to which to apply the content accelerator feature.

Note: For HTTP requests, the content accelerator feature can serve partial content in response to single byte-range requests.

The following figure illustrates the operations that the appliance performs when a client request arrives at a virtual server configured to use the content accelerator feature:



The process flow is as follows:

1. Client sends request.
2. NetScaler forwards the request to the server.
3. Server responds with the predefined size of the response (specified by the accumResSize parameter of the add ca action command).
4. NetScaler computes a hash of the response sent by the server.
5. NetScaler looks up the hash on the T2100 appliance.
6. A successful lookup indicates that the data is available and the T2100 appliance sends the data to the NetScaler.

   > **Note**:
   > When a lookup does not succeed, the NetScaler fetches all of the requested data from the server, and simultaneously serves the data to the client and updates the data on the T2100 appliance.
   > The T2100 appliance can be configured to specify the number of requests after which to cache the data.

7. NetScaler sends the response to the client.

Configure content accelerator

Before configuring the content accelerator feature, you must enable it on the NetScaler appliance.

You can configure the content accelerator feature to use one or multiple T2100 appliances. You must add each T2100 appliance as a service and bind these services to a load balancing virtual server that is dedicated to distributing the load between the configured T2100 appliances.

You must also configure a content accelerator action to lookup the data on the T2100 appliance. The action must also specify the T2100 load balancing virtual server and the size of data (in KB) to be fetched from the server to calculate the hash.

The action must be bound to a content accelerator policy that defines the traffic on which to perform content acceleration. The content accelerator policy must be bound to a content switching or load balancing virtual server that receives client traffic. Alternatively, you can bind the policy globally to be applicable to all virtual servers.

To configure content accelerator by using the command line interface

At the command prompt, do the following:

1. Enable the content accelerator feature.

   `enable ns feature ca`

2. Identify the T2100 appliances and add each as a service on the NetScaler appliance.

   `add service <name> <IPAddress> <serviceType> <port>`

   Example:

   > add service T2100-A 10.102.29.61 HTTP 30
   >
   > add service T2100-B 10.102.29.62 HTTP 40
   >
   > add service T2100-C 10.102.29.63 HTTP 50

> **Note**:
> The services must be of type HTTP only.
> 3. Create a load balancing virtual server for the T2100 appliances. Specify the token load balancing method and the rule shown in the following syntax.

```
'add lb vserver <name> <serviceType> <IPAddress> <port> -lbMethod TOKEN
    -rule"http.req.url.after_str(\"/lookup/\") alt http.req.url.path.
   SKIP(1).PREFIX(64)"'

**Example:**

`add lb vserver T2100-lbvserver HTTP 10.102.29.64 99 -lbMethod TOKEN -
   rule "http.req.url.after_str(\"/lookup/\") alt
http.req.url.path.SKIP(1).PREFIX(64)"` 4\.  Bind the T2100 services to
   the load balancing virtual server that you created for them.

`bind lb vserver <name> <serviceName>`

**Example:**
```

```

> bind lb vserver T2100-lbvserver T2100-A
> bind lb vserver T2100-lbvserver T2100-B
> bind lb vserver T2100-lbvserver T2100-C 5\.  Define a content
    accelerator action.

`add ca action <name> accumResSize <KBytes> -lbvserver <string> -
    type lookup`

Example
> add ca action ca_action1 -type lookup -lbvserver T2100-lbvserver -
    accumResSize 60 6\.  Define a content accelerator policy.

`add ca policy <name> -rule <expression> -action <name>`

**Example:** To create a content accelerator policy that caches all
    video formats.

`> add ca policy ca_mp4_pol -rule ns_video -action ca_action1`

where ns_video is a built-in expression.
```

7. Bind the content accelerator policy to either a virtual server that receives traffic or globally to the NetScaler system.

```
`bind lb vserver <name> -policyName <string>`

`bind cs vserver <name> -policyName <string>`

`bind ca global -policyName <string> -priority <num> -type <type>`

**Example:** To apply the content accelerator policy to a virtual
    server named "traf_rec"

`> bind lb vserver traf_rec -policyName ca_mp4_pol`

**Example:** To apply the content accelerator policy for all traffic
    reaching the NetScaler.

`> bind ca global -policyName ca_mp4_pol -priority 100 -type
    RES_DEFAULT`
```

8. Save the configuration.

```
`save ns config`
```

Configuring content accelerator by using the GUI

1. Navigate to **System** > **Settings** > **Configure Advanced Features** and select **Content Accelerator**.
2. Create a service for each of the T2100 appliances.
    a) Navigate to **Traffic Management** > **Load Balancing** > **Services**.
    b) Click **Add** and specify the relevant details. In the **Server** field, make sure you specify the IP address of the T2100 appliance. In the **Protocol** field select HTTP.
3. Create a virtual server and bind the T2100 services to it.
    a) Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
    b) Click **Add** and specify the relevant details.
    c) In the **Method and Persistence** tab, specify the Method as **Token**.
    d) In the **Policies** tab, specify the rule as http.req.url.after_str("/lookup/") alt http.req.url.path.SKIP(1).PR
    e) In the **Services** tab, select the T2100 services that you want to bind to the virtual server.
4. Create a content accelerator action.
    a) Navigate to **Optimization** > **Content Accelerator** > **Actions**.
    b) Specify the relevant details.
5. Create a content accelerator policy.
    a) Navigate to **Optimization** > **Content Accelerator** > **Policies**.
    b) Click **Add**, specify the policy rule, and associate the content accelerator action.
6. Bind the content accelerator policy globally or to a virtual server.
    a) Navigate to **Optimization** > **Content Accelerator**.
    b) Under the **Content Accelerator Policy Manager** [REQUEST] or Content Accelerator Policy Manager [RESPONSE] sections, bind the content accelerator policy globally or to a virtual server.

Media classification

September 21, 2020

Understanding the type of traffic in the network helps network administrators to manage bandwidth consumption for optimal network performance. The media classification mode monitors and displays the statistics of media traffic going through the NetScaler appliance.

With this mode enabled, a network administrators can collect stats showing the amount of data accessed, and the types of devices from which the media files have been accessed. The NetScaler appliance also supports byte-range requests in this mode.

Currently the NetScaler appliance can monitor and display statistics for the following media file types:

Media	File type
Microsoft Smooth Streaming	Video
Apple Live Streaming	Video
Audio Data Transport Stream (ADTS)	Audio
Advanced Audio Coding (AAC)	Audio
Flash Video (FLV)	Audio and Video
3GP	Audio and Video

The appliance can display stats for the following devices:

Device Platform	Device Type
iOS	iPad and iPod
Android	Mobiles and tablets
Laptop or Desktop	Windows laptop and desktop computers
Others	Other mobile devices (mobiles and tablets)

The network administrators can check the following stats counters to know the amount of data accessed through the NetScaler appliance for various media traffic types.

Media File Name	Stats Counter
Microsoft Smooth Streaming	mcmssmthstrmvid—This counter records the total number of Microsoft Smooth Streaming videos served by the NetScaler appliance; Mcmssmthstrvidpl—This counter records the total number of Microsoft Smooth Streaming video playlists served by the NetScaler appliance; Mcmssmthstrmvidbytes—This counter records the total number of data bytes served for Microsoft Smooth Streaming media traffic on the NetScaler appliance; Mcmssmthstrmplvidbytespl—This counter records the total number of Microsoft Smooth Streaming playlist bytes served by the NetScaler appliance.
Apple Live Streaming	mccaplelivestrmngvid—This counter records the total number of Apple Live Streaming videos served by the NetScaler appliance. Mccapplelivestrmngvidpl—This counter records the total number of Apple Live Streaming video playlists served by the NetScaler appliance. Mcapplelivestreamingvidbytes—This counter records the total number of data bytes served for Apple Live Streaming media traffic on the NetScaler appliance. Mcapplelivestreamingplaylistvidbytespl—This counter records the total number of Apple Live Playlist bytes served by the NetScaler appliance.
Audio Data Transport Stream (ADTS)	mcadtsaudio—This counter records the total number of ADTS audio clips served by the NetScaler appliance. Mcadtsaudiobytes—This counter records the total number of data bytes served for ADTS media traffic on the NetScaler appliance.

Media File Name	Stats Counter
Advanced Audio Coding (AAC)	Mcaacaudio—This counter records the total number of AAC audio clips served by the NetScaler appliance. Mcaacaudiobytes—This counter records the total number of data bytes served for AAC media traffic on the NetScaler appliance.
Flash Video (FLV)	Mcflvvid—This counter records the total number of flash videos served by the NetScaler appliance. Mcflvvidbytes—This counter records the total number of data bytes served for flash videos on the NetScaler appliance.
3GP	mc3gpvidbytes—This counter records the total number of data bytes served for 3GP media traffic on the NetScaler appliance.

The NetScaler appliance detects media file types by their signatures in the *initial body bytes* of the responses. For example, the initial body bytes for an mp4 file has the following signature in the response:

```
**....ftypmp42** ....isommp42....moov...lmvhd.....c.\!.c.\!..
```

The NetScaler appliance detects the client device type by the *user agent string* that the client device includes in the HTTP GET request. For example, a windows phone using a UC browser will have the following user agent string in the HTTP GET request:

```
User-Agent: **UCWEB**/2.0 (**Windows**; U; wds 8.10; en-US; HTC; 8X by HTC)
U2/1.0.0
```

Enable media classification

By default, media classification is disabled on the NetScaler appliance. You have to enable the mode before using it.

To enable media classification by using the command line interface

At the command prompt, type:

```
enable ns mode Mediaclassification
```

To enable media classification by using the GUI

Enable media classification on NetScaler appliance

Navigate to **System** > **Settings** > **Configure Modes** and select **Media Classification**.

To view media traffic statistics on the NetScaler appliance

Navigate to **Optimization** and click **Media Classification** to view the media traffic statistics.

Verify media classification statistics

You can view the media traffic statistics in the dashboard utility or using the command line interface. The dashboard utility displays summary and detailed statistics in a tabular and graphic format.

> **Note**
> For more information about statistics and charts, see the Dashboard help on your NetScaler appliance.

To View media classification statistics by using the command line interface

At the command prompt, type one of the following commands to display a summary of media classification statistics, display detailed statistics, or clear the display:

```
stat Mediaclassification
```

```
stat Mediaclassification -detail
```

```
stat Mediaclassification -clearstats
```

To view Media Classification statistics on the Dashboard

In the **Dashboard** utility, you can display the following types of media classification statistics:

1. Select **Media Classification** to display a summary of the media traffic statistics.
2. To display detailed media traffic statistics, click the **Details**.
3. To clear the media traffic statistics, click **Clear**.

Reputation

September 21, 2020

Citrix offers reputation based security. Using reputation assessment to determine the risk of processing requests, you can take actions such as blocking or dropping certain requests to improve the performance of your application.

The NetScaler IP reputation feature uses IP reputation checks to prevent Zero day attacks and provide protection against malicious sources associated with web attacks, phishing activity, or web scanning.

For additional details, see IP Reputation.

IP Reputation

September 22, 2020

IP reputation is an effective tool in identifying the IP address that is sending unwanted requests. Use the IP reputation list to preemptively reject requests that are coming from the IP with the bad reputation. For example, you can use this feature to optimize application firewall performance by filtering out the requests that you do not want to process. You can reset or drop the connection, or configure a responder policy to take a specific responder action.

Following are some examples of attacks that you can prevent by using IP Reputation:

- **Virus Infected personal computers** (home PCs) are the single biggest source of Spam on the internet. IP Reputation can identify the IP address that is sending unwanted requests. IP reputation can be especially useful for blocking large scale DDoS, DoS, or anomalous SYN flood attacks from known infected sources.
- **Centrally managed and automated botnet** attacks have gained popularity for stealing passwords, because it doesn't take long when hundreds of computers work together to crack your password. It is easy to launch botnet attacks to figure out passwords that use commonly used dictionary words.
- **Compromised web-server** Attacks are not as common because awareness and server security have increased, so hackers and spammers look for easier targets. There are still web servers and online forms that hackers can compromise and use to send spam (such as viruses and porn). Such activity is easier to detect and quickly shut down, or block with a reputation list such as SpamRats.
- **Windows Exploits** (such as Active IPs offering/distributing malware, shell code, rootkits, worms, or viruses).
- **Known spammers and hackers**.
- **Mass e-mail marketing campaigns**.
- **Phishing Proxies** (IP addresses hosting phishing sites, and other fraud such as ad click fraud or gaming fraud).
- **Anonymous proxies** (IPs providing proxy and anonymization services including The Onion Router aka TOR).

A NetScaler appliance uses **Webroot** as the service provider for the dynamically generated malicious IP database and the metadata for those IP addresses. Metadata might include geolocation details, threat category, threat count, and so on. The Webroot threat Intelligence engine receives real-time data from millions of sensors. It automatically and continuously captures, scans, analyses and scores the data, using advanced machine learning and behavioral analysis. Intelligence about a threat is continually updated.

When a threat is detected anywhere in the network, the IP address is flagged as malicious and all appliances connected to the network are immediately protected. The dynamic changes in the IP addresses are processed with high speed and accuracy by using advanced machine learning.

As stated in the IP reputation data sheet from Webroot, the Webroot's sensor network identifies many key IP threat types, including spam sources, Windows exploits, botnets, scanners, and others. (See the flow diagram on the data sheet)

The NetScaler appliance uses an `iprep` client process to get the database from Webroot. The `iprep` client uses the HTTP GET method to get the absolute IP list from Webroot for the first time. Later, it checks delta changes once every 5 minutes.

> **Important**
>
> - Make sure the NetScaler appliance has Internet access and DNS is configured before you use the IP Reputation feature.
> - To access the Webroot database, the NetScaler appliance must be able to connect to **api.bcti.brightcloud.com** on **port 443**. Each node in the HA or cluster deployment gets the database directly from the Webroot and must be able to access this FQDN (Fully Qualified Domain Name).
> - Webroot hosts its reputation database in AWS currently. Therefore, the NetScaler appliance must be able to resolve AWS domains for downloading the reputation db. Also, the firewall must be open for AWS domains.
> - NetScaler appliance must be able to connect to **wiprep-rtu.s3-us-west-2.amazonaws.com** on **port 443** to obtain IP data from AWS.
>
> **Note:**
>
> Each packet engine requires at least 4 GB to function properly when the IP Reputation feature is enabled.

PI Expressions: The IP Reputation feature can be configured by using PI expressions (NetScaler default syntax expressions) in the policies bound to supported modules such as application firewall and responder. Following are two examples showing expressions that can be used to detect whether the client IP address is malicious

1. **CLIENT.IP.SRC.IPREP_IS_MALICIOUS**: This expression evaluates to TRUE if the client is included in the malicious IP list.
2. **CLIENT.IP.SRC.IPREP_THREAT_CATEGORY (CATEGORY)**: This expression evaluates to TRUE if the client IP is malicious IP and is in the specified threat category.

Following are the possible values for the threat category:

SPAM_SOURCES, WINDOWS_EXPLOITS, WEB_ATTACKS, BOTNETS,
SCANNERS, DOS, REPUTATION, PHISHING, PROXY, NETWORK,
CLOUD_PROVIDERS, MOBILE_THREATS.

> **Note**
>
> The IP reputation feature can check both source and destination IP addresses. It can also detect malicious IPs in the header. If the PI Expression in a policy can identify the IP address, the IP reputation check can determine whether it is malicious.

IPRep log message. The `/var/log/iprep.log` file contains useful messages that capture information about communication with the Webroot database. The information can be about the credentials used during Webroot communication, failure to connect with Webroot, information included in an update (such as the number of IP addresses in the database).

Creating a blacklist or whitelist of IPs using a policy data set. You can maintain an allow list to allow access to specific IP addresses that are blacklisted in the Webroot database. You can also create a customized block list of IP addresses to supplement the Webroot reputation check. These lists can be created by using a policy **data set**. A data set is a specialized form of pattern set that is ideally suited for IPv4 address matching. To use data sets, first create the data set and bind IPv4 addresses to it. Then, when you configure a policy for comparing a string in a packet, use an appropriate operator and pass the name of the pattern set or data set as an argument.

To create an allow list of addresses to treat as exceptions during IP reputation evaluation:

- Configure the policy so that the PI expression evaluates to False even if an address in the allow list is listed as malicious by Webroot (or any service provider).

Enabling or disabling IP reputation. IP reputation is a part of the general reputation feature, which is license based. When you enable or disable the reputation feature, it enables or disables IP Reputation.

General procedure. Deploying IP reputation involves the following tasks

- Verify that the license installed on the NetScaler appliance has IP reputation support. Platinum and standalone application firewall licenses support the IP reputation feature.
- Enable the IP reputation and application firewall features.
- Add an application firewall profile.
- Add an application firewall policy using the PI expressions to identify the malicious IP addresses in the IP Reputation database.
- Bind the application firewall policy to an appropriate bind point.
- Verify that any request received from a malicious address gets logged in the ns.log file to show that the request was processed as specified in the profile.

Using the Command Line to Configure the IP Reputation Feature

To enable IP reputation using the CLI, you can use the following commands:

```
enable feature reputation

```

```
disable feature reputation
<!--NeedCopy-->
```

The following examples show how you can add an application firewall policy using the PI expression to identify malicious addresses. You can use the built-in profiles, or add profile, or configure an existing profile to invoke the desired action when a request matches a policy match.

Examples 3 and 4 show how to create a policy dataset to generate a Black (to be blocked) or White (to be allowed) list of IP addresses.

Example 1:

The following command creates a policy that identifies malicious IP addresses and block the request if a match is triggered:

```
add appfw policy pol1 CLIENT.IP.SRC.IPREP_IS_MALICIOUS APPFW_BLOCK
<!--NeedCopy-->
```

Example 2:

The following command creates a policy that uses the reputation service to check the client IP address in the `X-Forwarded-For` header and reset the connection if a match is triggered.

```
add appfw policy pol1 "HTTP.REQ.HEADER(\"X-Forwarded-For\").
    TYPECAST_IP_ADDRESS_AT.IPREP_IS_MALICIOUS" APPFW_RESET
<!--NeedCopy-->
```

Example 3:

The following example shows how to add a list to add exceptions that allow specified IP addresses:

```
add policy dataset Allow_list ipv4

bind policy dataset Allow_list 10.217.25.17 -index 1

bind policy dataset Allow_list 10.217.25.18 -index 2
<!--NeedCopy-->
```

Example 4:

The following example shows how to add the customized list to flag specified IP addresses as malicious:

```
add policy dataset Block_List ipv4

bind policy dataset Block_List 10.217.31.48 -index 1

bind policy dataset Block_List 10.217.25.19 -index 2
<!--NeedCopy-->
```

Example 5:

The following example shows a policy expression to block the client IP in the following conditions:

- It matches an IP address configured in the customized Block_list1 (example 4)
- It matches an IP address listed in the Webroot database unless relaxed by inclusion in the Allow_list1 (example 3).).

```
add appfw policy "Ip_Rep_Policy" "((CLIENT.IP.SRC.IPREP_IS_MALICIOUS || CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Block_List\")) && ! (CLIENT.IP.SRC.TYPECAST_TEXT_T.CONTAINS_ANY(\"Allow_List\")))" APPFW_BLOCK
<!--NeedCopy-->
```

Using Proxy server:

If the appliance does not have direct access to the Internet and is connected to a proxy, configure the IP Reputation client to send requests to the proxy.

```
set reputation settings  - proxyServer <proxy server ip> -proxyPort <proxy server port>
<!--NeedCopy-->
```

Example:

```
set reputation settings proxyServer 10.102.30.112 proxyPort 3128

set reputation settings  - proxyServer testproxy.citrite.net   - proxyPort 3128

unset reputation settings  - proxyserver  - proxyport

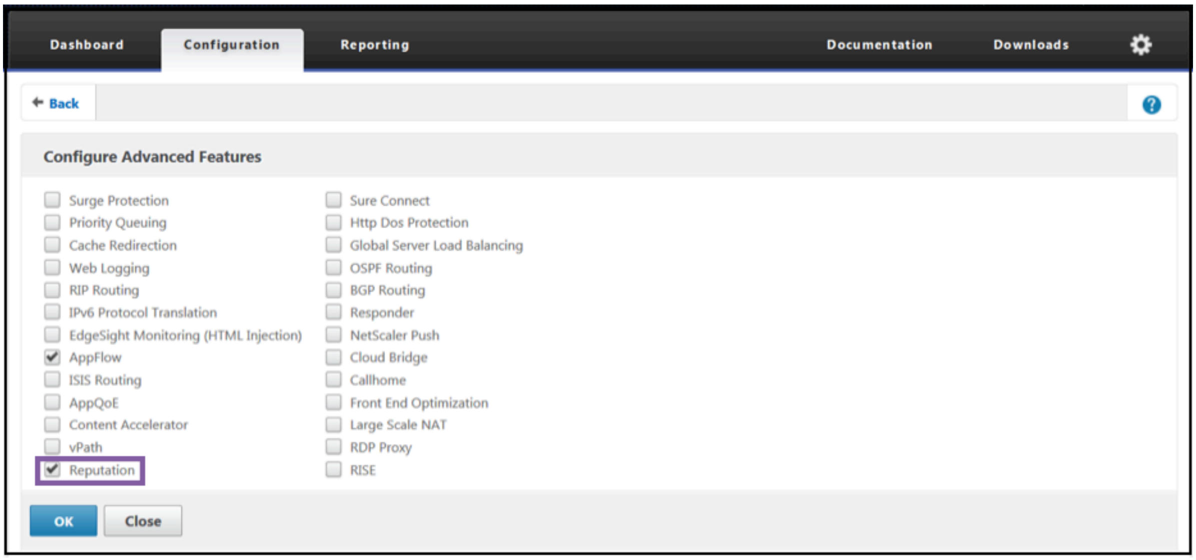
sh reputation settings
<!--NeedCopy-->
```

> **Note**
>
> The Proxy Server IP can be an IP address or a fully qualified domain name (FQDN).

Using the GUI to Configure the IP Reputation Feature

To enable reputation (which enables IP reputation) feature in GUI:

Navigate to the **System** > **Settings**. In the **Modes and Features** section, click the link to access the **Configure Advanced Features** pane and enable the **Reputation** check box. Click **OK**.

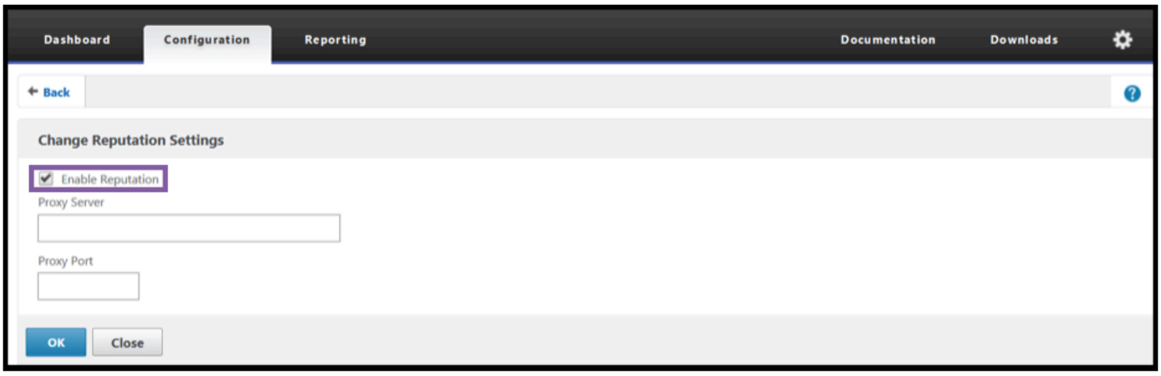

Dashboard
Configuration
Reporting
Documentation
Downloads
Back
OK
Close

Configure Advanced Features

☐ Surge Protection
☐ Priority Queuing
☐ Cache Redirection
☐ Web Logging
☐ RIP Routing
☐ IPv6 Protocol Translation
☐ EdgeSight Monitoring (HTML Injection)
☑ AppFlow
☐ ISIS Routing
☐ AppQoE
☐ Content Accelerator
☐ vPath
☑ Reputation
☐ Sure Connect
☐ Http Dos Protection
☐ Global Server Load Balancing
☐ OSPF Routing
☐ BGP Routing
☐ Responder
☐ NetScaler Push
☐ Cloud Bridge
☐ Callhome
☐ Front End Optimization
☐ Large Scale NAT
☐ RDP Proxy
☐ RISE

To configure a proxy server by using the configuration utility:

1. On the configuration tab, navigate to **Security** > **Reputation**. Under **Settings**, click **Change Reputation Settings** to configure a proxy server. You can also enable or disable the reputation feature. **Proxy Server** can be an IP address or a fully qualified domain name (FQDN). **Proxy port** accepts values between [1–65535].


Dashboard
Configuration
Reporting
Documentation
Downloads
Back
OK
Close

Change Reputation Settings

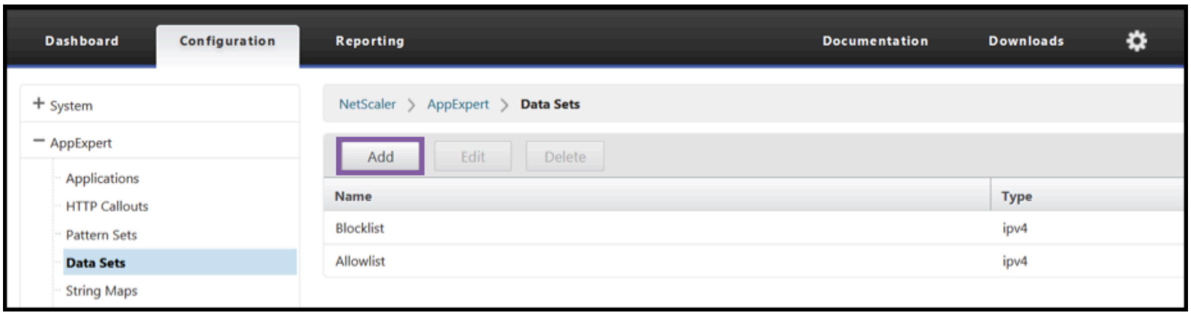
☑ Enable Reputation

Proxy Server

Proxy Port

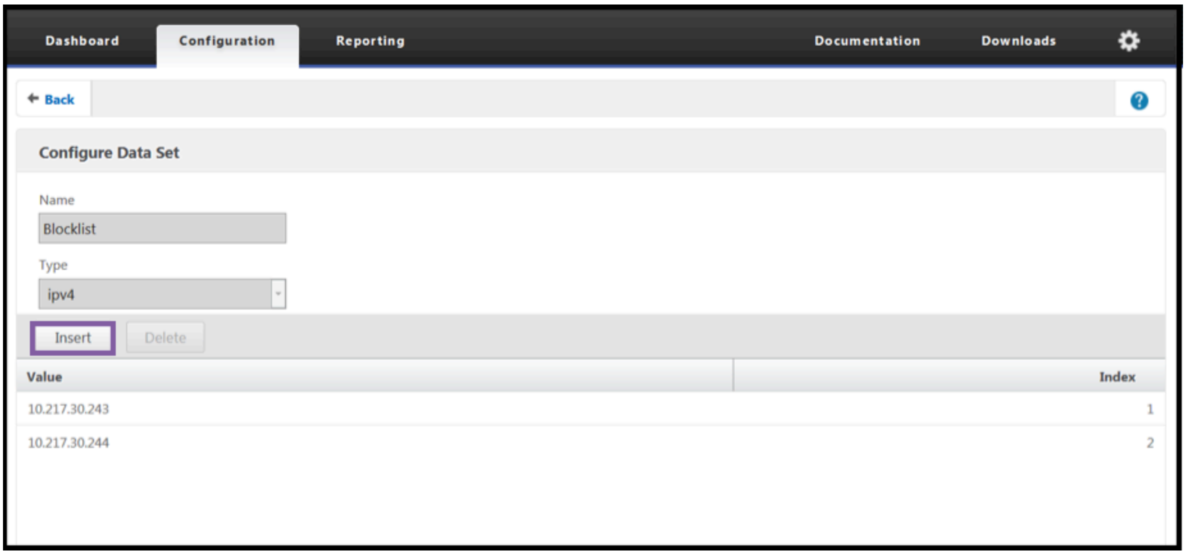
To use a dataset to create a whitelist (list of safe IPs) and blacklist (list of unsafe IPs) of Client IP addresses:

- On the **Configuration** tab, navigate to **AppExpert** > **Data Sets**.
- Click **Add**.


Dashboard
Configuration
Reporting
Documentation
Downloads
+ System
− AppExpert
Applications
HTTP Callouts
Pattern Sets
Data Sets
String Maps
NetScaler > AppExpert > Data Sets
Add
Edit
Delete

Name	Type
Blocklist	ipv4
Allowlist	ipv4

- In the **Create Data Set** (or **Configure Data set**) pane, provide a meaningful name for the list of the IP addresses. The name must reflect the purpose of the list.
- Select **Type** as **IPv4**.
- Click **Insert** to add an entry.


Dashboard
Configuration
Reporting
Documentation
Downloads
Back
Configure Data Set
Insert
Delete

Name: Blocklist

Type: ipv4

Value	Index
10.217.30.243	1
10.217.30.244	2

- In the Configure Policy dataset binding pane, add an IPv4 format IP address in the Value input box.
- Provide an index.
- Add a comment that explains the purpose of the list. This step is optional, but is recommended because a descriptive comment is helpful in managing the list.

Similarly, you can create a Block_list and add the IP addresses that are to be considered malicious.

Also see, Pattern sets and data sets for more details regarding using data sets and configuring default syntax policy expressions.

To configure an application firewall policy by using the configuration utility:

1. On the **Configuration** tab, navigate to **Security > Web App Firewall > Policies > Firewall**. Click **Add** to add a policy using the PI expressions to use IP reputation.

You can also use the Expression editor to build your own policy expression. The list shows preconfigured options that are useful for configuring an expression using the threat categories.

Highlights

- Quickly and accurately stop bad traffic at the network's edge from known malicious IP addresses posing different types of threats. You can block the request without parsing the body.
- Dynamically configure IP reputation functionality for multiple applications.
- Secure your network against data breach without a performance penalty, and consolidate protections onto a single services fabric using fast and easy deployments.
- You can do IP Reputation checks on source and destination IPs.
- You can also inspect the headers to detect malicious IPs.
- IP reputation check is supported in both forward proxy and reverse proxy deployments.
- The `iprep` process connects with Webroot and updates the database every 5 minutes.
- Each node in the High Availability (HA) or Cluster deployment gets the database from Webroot.
- The IP reputation data is shared across all partitions in admin-partition deployments.
- You can use an AppExpert data set to create lists of IP addresses to add exceptions for IPs blacklisted in the Webroot database. You can also create your own customized block list to designate specific IPs as malicious.
- The iprep.db file is created in the /var/nslog/iprep folder. Once created, it is not deleted even if the feature is disabled.
- When the reputation feature is enabled, the NetScaler Webroot database is downloaded. After that, it is updated every 5 minutes.
- The Webroot database major version is version: 1.
- The minor version gets updated every day. The update version is incremented after every 5 minutes and is reset back to 1 when the minor version is incremented.
- PI expressions enable you to use IP reputation with other features, such as responder and rewrite.
- The IP addresses in the database are in decimal notation.

Debugging Tips

- If you cannot see the reputation feature in the GUI, verify that you have the right license
- Monitor the messages in `/var/log/iprep.log` for debugging.
- **Webroot connectivity**: If you see the `ns iprep: Not able to connect/resolve Webroot` message, make sure that the appliance has internet access and DNS is configured.
- **Proxy server**: If you see the `ns iprep: iprep_curl_download: 88 curl_easy_perform failed. Error code: 5 Err msg:couldnt resolve proxy name` message, make sure that the proxy server configuration is accurate.
- **IP Reputation feature not working**: The `iprep` process takes about five minutes to start after you enable the reputation feature. The IP reputation feature might not work for that duration.
- **Database download**: If the IP DB data download is failing after enabling the IP Reputation feature, the following error is seen in the logs.

```
iprep: iprep_curl_download:86 curl_easy_perform failed. Error code:7 Err
msg:Couldn't connect to server
```

Solution: Allow the out-bound traffic to the following URLs or configure a proxy to resolve the issue.

```
localdb-ip-daily.brightcloud.com:443
localdb-ip-rtu.brightcloud.com:443
api.bcti.brightcloud.com:443
<!--NeedCopy-->
```

SSL offload and acceleration

September 21, 2020

A NetScaler appliance configured for SSL acceleration transparently accelerates SSL transactions by offloading SSL processing from the server. To configure SSL offloading, you configure a virtual server to intercept and process SSL transactions, and send the decrypted traffic to the server (unless you configure end-to-end encryption, in which case the traffic is re-encrypted). Upon receiving the response from the server, the appliance completes the secure transaction with the client. From the client's perspective, the transaction seems to be directly with the server. A NetScaler configured for SSL acceleration also performs other configured functions, such as load balancing.

Configuring SSL offloading requires an SSL certificate and key pair, which you must obtain if you do not already have an SSL certificate. Other SSL-related tasks that you might need to perform include managing certificates, managing certificate revocation lists, configuring client authentication, and managing SSL actions and policies.

A non-FIPS NetScaler appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as a hardware security module (HSM). Only the MPX 9700/10500/12500/15500 appliances support a FIPS card, so other NetScaler models cannot be equipped with an HSM.

Beginning with release 10.5, build 52.1115.e, all NetScaler appliances that do not support a FIPS card (including virtual appliances) support the Thales nShield® Connect external HSM. (MPX 9700/10500/12500/15500 appliances do not support an external HSM.)

Note: FIPS-related options for some of the SSL configuration procedures described in this document are specific to a FIPS-enabled NetScaler appliance.

SSL offloading configuration

September 21, 2020

To configure SSL offloading, you must enable SSL processing on the NetScaler appliance and configure an SSL based virtual server that will intercept SSL traffic, decrypt the traffic, and forward it to a service that is bound to the virtual server. To secure time-sensitive traffic, such as media streaming, you can configure a DTLS virtual server. To enable SSL offloading, you must import a valid certificate and key and bind the pair to the virtual server.

Enable SSL

To process SSL traffic, you must enable SSL processing. You can configure SSL based entities, such as virtual servers and services, without enabling SSL processing, but they will not work until SSL processing is enabled.

Enable SSL processing by using the CLI

At the command prompt, type:

```
enable ns feature ssl

show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature SSL
 Done
show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   OFF
 2)     Surge Protection               SP                   ON
 3)     Load Balancing                 LB                   ON
 .
 .
 .
 9)     SSL Offloading                 SSL                  ON
 .
 .
 .
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

Enable SSL processing by using the GUI

Navigate to **System > Settings** and, in the Modes and Features group, select **Configure Basic Features**, and select **SSL Offloading**.

Configure services

On the NetScaler appliance, a service represents a physical server or an application on a physical server. Once configured, services are in the disabled state until the appliance can reach the physical server on the network and monitor its status.

Add a service by using the CLI

At the command prompt, type the following commands to add a service and verify the configuration:

```
add service <name> (<IP> | <serverName>) <serviceType> <port>
show service <serviceName>
<!--NeedCopy-->
```

Example:

```
add service sslsvc 198.51.100.225 SSL 443

Done

sh ssl service sslsvc

            Advanced SSL configuration for Back-end SSL Service sslsvc:

            DH: DISABLED

            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA: DISABLED

            Session Reuse: ENABLED          Timeout: 300 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            ClearText Port: 0

            Server Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            OCSP Stapling: DISABLED

            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED

            Send Close-Notify: YES

            Strict Sig-Digest Check: DISABLED

            Zero RTT Early Data: ???

            DHE Key Exchange With PSK: ???

            Tickets Per Authentication Context: ???



            ECC Curve: P_256, P_384, P_224, P_521


    1)      Cipher Name: DEFAULT_BACKEND

            Description: Default cipher list for Backend SSL session

Done
<!--NeedCopy-->
```

Modify or remove a service by using the CLI

To modify a service, use the set service command, which is just like using the add service command, except that you enter the name of an existing service. To remove a service, use the rm service command, which accepts only the `<name>` argument.

Configure a service by using the GUI

Navigate to **Traffic Management > Load Balancing > Services**, create a service, and specify the protocol as SSL.

SSL virtual server configuration

Secure sessions require establishing a connection between the client and an SSL-based virtual server on the NetScaler appliance. The SSL virtual server intercepts SSL traffic, decrypts it and processes it before sending it to services that are bound to the virtual server.

> **Note:**
>
> The SSL virtual server is marked as down on the NetScaler appliance until a valid certificate / key pair and at least one service are bound to it. An SSL based virtual server is a load balancing virtual server of protocol type SSL or SSL_TCP. The load balancing feature must be enabled on the NetScaler appliance.

Add an SSL-based virtual server by using the CLI

At the command prompt, type the following commands to create an SSL-based virtual server and verify the configuration:

```
add lb vserver <name> (serviceType) <IPAddress> <port>
show ssl vserver <name>
<!--NeedCopy-->
```

Example:

```
add lb vserver sslvs SSL 192.0.2.240 443
Done

sh ssl vserver sslvs

      Advanced SSL configuration for VServer sslvs:

      DH: DISABLED

      DH Private-Key Exponent Size Limit: DISABLED  Ephemeral RSA: ENABLED     Refresh Count: 0

      Session Reuse: ENABLED     Timeout: 120 seconds

      Cipher Redirect: DISABLED
```

```

            SSLv2 Redirect: DISABLED

            ClearText Port: 0

            Client Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            OCSP Stapling: DISABLED

            HSTS: DISABLED

            HSTS IncludeSubDomains: NO

            HSTS Max-Age: 0

            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED

            Push Encryption Trigger: Always

            Send Close-Notify: YES

            Strict Sig-Digest Check: DISABLED

            Zero RTT Early Data: DISABLED

            DHE Key Exchange With PSK: NO

            Tickets Per Authentication Context: 1

            ECC Curve: P_256, P_384, P_224, P_521


    1)      Cipher Name: DEFAULT

            Description: Default cipher list with encryption strength >= 128bit

Done
```

```
<!--NeedCopy-->
```

Modify or remove an SSL-based virtual server by using the CLI

To modify the load balancing properties of an SSL virtual server, use the `set lb vserver` command, which is just like using the `add lb vserver` command, except that you enter the name of an existing vserver. To modify the SSL properties of an SSL-based virtual server, use the `set ssl vserver` command. For more information, see the "SSL virtual server parameters" section later in this page.

To remove an SSL virtual server, use the `rm lb vserver` command, which accepts only the `<name>` argument.

Configure an SSL-based virtual server by using the GUI

Navigate to **Traffic Management > Load Balancing > Virtual Servers**, create a virtual server, and specify the protocol as SSL.

Bind services to the SSL virtual server

For the NetScaler appliance to forward decrypted SSL data to servers in the network, services representing these physical servers must be bound to the virtual server that receives the SSL data.

Because the link between the NetScaler appliance and the physical server is typically secure, data transfer between the appliance and the physical server does not have to be encrypted. However, you can provide end-to end-encryption by encrypting data transfer between the appliance and the server. For details, see Configure SSL offloading with end-to-end encryption.

Note: Enable the load balancing feature on the NetScaler appliance before you bind services to the SSL based virtual server.

Bind a service to a virtual server by using the CLI

At the command prompt, type the following commands to bind the service to the virtual server and verify the configuration:

```
bind lb vserver <name> <serviceName>
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind lb vserver sslvs sslsvc

```

```
Done

sh lb vserver sslvs

            sslvs (192.0.2.240:443) - SSL       Type: ADDRESS

            State: DOWN[Certkey not bound]

            Last state change was at Wed May  2 11:43:04 2018

            Time since last state change: 0 days, 00:13:21.150

            Effective State: DOWN

            Client Idle Timeout: 180 sec

            Down state flush: ENABLED

            Disable Primary Vserver On Down : DISABLED

            Appflow logging: ENABLED

            No. of Bound Services :  1 (Total)       0 (Active)

            Configured Method: LEASTCONNECTION      BackupMethod: ROUNDROBIN

            Mode: IP

            Persistence: NONE

            Vserver IP and Port insertion: OFF

            Push: DISABLED  Push VServer:

            Push Multi Clients: NO

            Push Label Rule: none

            L2Conn: OFF

            Skip Persistency: None

            Listen Policy: NONE

            IcmpResponse: PASSIVE

            RHIstate: PASSIVE

            New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0

            Mac mode Retain Vlan: DISABLED

            DBS_LB: DISABLED

            Process Local: DISABLED

            Traffic Domain: 0

            TROFS Persistence honored: ENABLED

            Retain Connections on Cluster: NO

    1) sslsvc (198.51.100.225: 443) - SSL State: DOWN       Weight: 1

Done
<!--NeedCopy-->
```

Unbind a service from a virtual server by using the CLI

At the command prompt, type the following command:

```
unbind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
unbind lb vserver sslvs sslsvc

Done
<!--NeedCopy-->
```

Bind a service to a virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open a virtual server, and click in the Services section to bind a service to the virtual server.

Configure an SNI virtual server for secure hosting of multiple sites

Virtual hosting is used by Web servers to host more than one domain name with the same IP address. The appliance supports hosting of multiple secure domains by offloading SSL processing from the Web servers using transparent SSL services or virtual server-based SSL offloading. However, when multiple websites are hosted on the same virtual server, the SSL handshake is completed before the expected host name is sent to the virtual server. As a result, the appliance cannot determine which certificate to present to the client after a connection is established. This problem is resolved by enabling Server Name Indication (SNI) on the virtual server. SNI is a Transport Layer Security (TLS) extension used by the client to provide the host name during handshake initiation. The NetScaler appliance compares this host name to the common name and, if it does not match, compares it to the subject alternative name (SAN). If the name matches, the appliance presents the corresponding certificate to the client.

A wildcard SSL Certificate helps enable SSL encryption on multiple subdomains if the domains are controlled by the same organization and share the second-level domain name. For example, a wildcard certificate issued to a sports network using the common name "*.sports.net" can be used to secure domains, such as "login.sports.net" and "help.sports.net" but not "login.ftp.sports.net."

Notes:

- On a NetScaler appliance, only domain name, URL, and email ID DNS entries in the SAN field are compared.
- If an existing certificate-key pair (for example, certkey A) is bound to an SSL virtual server, and a second certificate-key pair (for example, certkey B) is added using SNI and bound to the same virtual server, certkey A must also be configured as `-SNICert`. If a Client Hello message includes an SNI extension with the domain name of certkey A and the above configuration is not in place, TLS connections to the domains in (certkey A) will not work.

You can bind multiple server certificates to a single SSL virtual server or transparent service using the -SNICert option. These certificates are issued by the virtual server or service if SNI is enabled on the virtual server or service. You can enable SNI at any time.

Bind multiple server certificates to a single SSL virtual server by using the CLI

At the command prompt, type the following commands to configure SNI and verify the configuration:

```
set ssl vserver <vServerName>@ [-SNIEnable ( ENABLED | DISABLED )]

bind ssl vserver <vServerName>@ -certkeyName <string> -SNICert

show ssl vserver <vServerName>
<!--NeedCopy-->
```

To bind multiple server certificates to a transparent service by using the CLI, replace `vserver` with `service` and `vservername` with `servicename` in the preceding commands.

Note: Create the SSL service with `-clearTextPort 80` option.

Bind multiple server certificates to a single SSL virtual server by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**.
2. Open an SSL virtual server and, in Certificates, click **Server Certificate**.
3. Add a new certificate or select a certificate from the list, and select **Server Certificate for SNI**.
4. In **Advanced Settings**, click **SSL Parameters**.
5. Select **SNI Enable**.

Support for SNI on the back-end service

> **Note:**
>
> SNI is not supported on a DTLS back-end service.

The NetScaler appliance supports Server Name Indication (SNI) at the back end. That is, the common name is sent as the server name in the client hello to the back-end server for successful completion of the handshake. In addition to helping meet federal system integrator customer security requirements, this enhancement provides the advantage of using only one port instead of opening hundreds of different IP addresses and ports on a firewall.

Federal system integrator customer security requirements include support for Active Directory Federation Services (ADFS) 3.0 in 2012R2 and WAP servers. This requires supporting SNI at the back end on a NetScaler appliance.

> **Note:**
>
> For SNI to work, the server name in the client hello must match the host name configured on the back-end service that is bound to an SSL virtual server. For example, if the host name of the back-end server is www.mail.example.com, the SNI-enabled back-end service must be configured with the server name as https://www.mail.example.com, and this host name must match the server name in the client hello.

Configure SNI on the back-end service by using the CLI

At the command prompt, type:

```
add service <name>  <IP>  <serviceType>  <port>

add lb vserver <name>  <IPAddress> <serviceType>  <port>

```

```
bind lb vserver <name> <serviceName>

set ssl service <serviceName> -SNIEnable ENABLED -commonName <string>
<!--NeedCopy-->
```

Example:

```
add service service_ssl 198.51.100.100 SSL 443

add lb vserver ssl-vs 203.0.113.200 SSL 443

bind lb vserver ssl-vs service_ssl

set ssl service service_ssl -SNIEnable ENABLED  - commonName www.example.com
<!--NeedCopy-->
```

Configure SNI on the back-end service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
2. Select an SSL service, and in **Advanced Settings**, select **SSL Parameters**.
3. Select **SNI Enable**.

Bind a secure monitor to an SNI-enabled back-end service

You can also bind secure monitors of type HTTP-ECV or TCP-ECV to the back-end services that support SNI. To do this, the custom header in the monitor must be set to the server name that is sent as the SNI extension in the client hello.

Configure and bind a secure monitor to an SNI-enabled back-end service by using the CLI

At the command prompt, type:

```
add lb monitor <monitorName> <type>

set lb monitor <monitorName> <type> -customHeaders <string>

bind service <name> -monitorName <string>
<!--NeedCopy-->
```

Example:

```
add monitor https-ecv-mon http-ecv
Done
set monitor https-ecv-mon HTTP-ECV -customHeaders "Host: example.com\r\
   n"
Done
bind service ssl_service  - monitorName https-ecv
<!--NeedCopy-->
```

Configure and bind a secure monitor to an SNI-enabled back-end service by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Monitor**.
2. Add a monitor of type **HTTP-ECV** or **TCP-ECV**, and specify a **Custom Header**.
3. Click **Create**.
4. Navigate to **Traffic Management** > **Load Balancing** > **Services**.
5. Select an SSL service and click **Edit**.
6. In **Monitors**, click **Add Binding**, select the monitor created in step 3, and click **Bind**.

Add or update a certificate-key pair

Caution: If while adding a certificate-key pair, you add a certificate file with the same name as an existing certificate file but with different content, the original certificate file is overwritten with no warning. This might cause issues after the appliance is restarted because the original certificate file is no longer available in the /nsconfig/ssl directory.

> **Note:**
>
> If you don't have an existing certificate and key, see Create a certificate.
>
> To create an ECDSA certificate-key pair, click Create an ECDSA certificate-key pair.

For any SSL transaction, the server needs a valid certificate and the corresponding private and public key pair. The SSL data is encrypted with the server's public key, which is available through the server's certificate. Decryption requires the corresponding private key.

Because the NetScaler appliance offloads SSL transactions from the server, the server's certificate and private key must be present on the appliance, and the certificate must be paired with its corresponding private key. This certificate-key pair must then be bound to the virtual server that processes the SSL transactions.

> **Note:**
>
> From release 11.0, the default certificate on a NetScaler appliance is 2048 bits. In earlier builds, the default certificate was 512 bits or 1024 bits. After upgrading to release 11.0, you must delete

> all your old certificate-key pairs starting with "ns-", and then restart the appliance to automatically generate a 2048-bit default certificate.
>
> Both the certificate and the key must be in local storage on the NetScaler appliance before they can be added to the appliance. If your certificate or key file is not on the appliance, upload it to the appliance before you create the pair.

Important: Certificates and keys are stored in the /nsconfig/ssl directory by default. If your certificates or keys are stored in any other location, you must provide the absolute path to the files on the NetScaler appliance. The NetScaler FIPS appliances do not support external keys (non-FIPS keys). On a FIPS appliance, you cannot load keys from a local storage device such as a hard disk or flash memory. The FIPS keys must be present in the Hardware Security Module (HSM) of the appliance.

On a NetScaler MPX appliance and a NetScaler FIPS appliance, only RSA private keys are supported. On a VPX virtual appliance, both RSA and DSA private keys are supported. On an SDX appliance if SSL chips are assigned to an instance, then only RSA private keys are supported. However, if SSL chips are not assigned to an instance, then both RSA and DSA private keys are supported. In all the cases, you can bind a CA certificate with either RSA or DSA keys.

Set the notification period and enable the expiry monitor to issue a prompt before the certificate expires.

The NetScaler appliance supports the following input formats of the certificate and the private-key files:

- PEM - Privacy Enhanced Mail
- DER - Distinguished Encoding Rule
- PFX - Personal Information Exchange

The format is automatically detected by the software. Therefore, you are no longer required to specify the format in the inform parameter. If you do specify the format (correct or incorrect), it is ignored by the software. The format of the certificate and the key file must be the same.

> **Note:**
>
> A certificate must be signed by using one of the following hash algorithms:
>
> - MD5
> - SHA-1
> - SHA-224
> - SHA-256
> - SHA-384 (supported only on the front end)
> - SHA-512 (supported only on the front end)

An MPX appliance supports certificates of 512 or more bits, up to the following sizes:

- 4096-bit server certificate on the virtual server

- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back-end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

A VPX virtual appliance supports certificates of 512 or more bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 2048-bit certificate on the back-end server
- 2048-bit client certificate (if client authentication is enabled on the virtual server)

> **Note**
>
> A NetScaler SDX appliance supports certificates of 512 or more bits. Each NetScaler VPX instance hosted on the appliance supports the certificate sizes listed above for a VPX virtual appliance. However, if an SSL chip is assigned to an instance, that instance supports the certificate sizes supported by an MPX appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type the following commands to add a certificate-key pair and verify the configuration:

```
add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password ]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <positive_integer>]]

show ssl certKey [<certkeyName>]
<!--NeedCopy-->
```

Example:

```
add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -password ssl -expiryMonitor ENABLED -notificationPeriod 30
 Done
Note: For FIPS appliances, replace -key with -fipskey
show ssl certKey sslckey
        Name: sslckey           Status: Valid,   Days to expiration:8418
        Version: 3
        Serial Number: 01
        Signature Algorithm: md5WithRSAEncryption
        Issuer:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.root.com
        Validity
                Not Before: Jul 15 02:25:01 2005 GMT
                Not After : Nov 30 02:25:01 2032 GMT
        Subject:  C=US,ST=SJ,L=SJ,O=NS,OU=NSSSL,CN=www.server.com
        Public Key Algorithm: rsaEncryption
        Public Key size: 2048
Done
<!--NeedCopy-->
```

Update or remove a certificate-key pair by using the CLI

To modify the expiry monitor or notification period in a certificate-key pair, use the `set ssl certkey` command. To replace the certificate or key in a certificate-key pair, use the `update ssl certkey` command. The `update ssl certkey` command has an additional parameter for overriding the domain check. For both commands, enter the name of an existing certificate-key pair. To remove an SSL certificate-key pair, use the `rm ssl certkey` command, which accepts only the `<certkeyName>` argument.

Example:

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED )
       [-notificationPeriod <positive_integer>]]

update ssl certKey <certkeyName> [-cert <string> [-password]] [-key
       <string> | -fipsKey <string>] [-inform <inform>] [-noDomainCheck ]
<!--NeedCopy-->
```

Add or update a certificate-key pair by using the GUI

Navigate to **Traffic Management** > **SSL** > **Certificates**, and configure a certificate-key pair.

Bind the certificate-key pair to the SSL virtual server

An SSL certificate is an integral element of the SSL encryption and decryption process. The certificate is used during an SSL handshake to establish the identity of the SSL server.

The certificate being used for processing SSL transactions must be bound to the virtual server that receives the SSL data. If you have multiple virtual servers receiving SSL data, a valid certificate-key pair must be bound to each of them.

You can use a valid, existing SSL certificate that you have uploaded to the NetScaler appliance. As an alternative for testing purposes, you can create your own SSL certificate on the appliance. Intermediate certificates created by using a FIPS key on the appliance cannot be bound to an SSL virtual server.

As a part of the SSL handshake, in the certificate request message during client authentication, the server lists the distinguished names (DNs) of all the certificate authorities (CAs) bound to the server from which it accepts a client certificate. If you do not want the DN name of a specific CA certificate to be sent to the SSL client, set the skipCA flag. This setting indicates that the particular CA certificate's distinguished name should not be sent to the SSL client.

For details on how to create your own certificate, see Managing Certificates.

> **Note:**
>
> Citrix recommends that you use only valid SSL certificates issued by a trusted certificate authority.

Bind an SSL certificate-key pair to a virtual server by using the CLI

At the command prompt, type the following commands to bind an SSL certificate-key pair to a virtual server and verify the configuration:

```
bind ssl vserver <vServerName> -certkeyName <certificate-KeyPairName> -
    CA -skipCAName
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
bind ssl vs vs1 -certkeyName cert2 -CA -skipCAName
Done
sh ssl vs vs1

 Advanced SSL configuration for VServer vs1:

 DH: DISABLED

 Ephemeral RSA: ENABLED Refresh Count: 0

 Session Reuse: ENABLED Timeout: 120 seconds

 Cipher Redirect: DISABLED

 SSLv2 Redirect: DISABLED

 ClearText Port: 0

 Client Auth: DISABLED

 SSL Redirect: DISABLED

 Non FIPS Ciphers: DISABLED

 SNI: DISABLED

 OCSP Stapling: DISABLED

 HSTS: DISABLED

 IncludeSubDomains: NO

 HSTS Max-Age: 0

 SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

 Push Encryption Trigger: Always

 Send Close-Notify: YES

 Strict Sig-Digest Check: DISABLED

ECC Curve: P_256, P_384, P_224, P_521

 1) CertKey Name: cert1 CA Certificate OCSPCheck: Optional CA_Name Sent
 2) CertKey Name: cert2 CA Certificate OCSPCheck: Optional CA_Name Skipped
 1) Cipher Name: DEFAULT

Description: Default cipher list with encryption strength >= 128bit
Done
<!--NeedCopy-->
```

Unbind an SSL certificate-key pair from a virtual server by using the CLI

If you try to unbind a certificate-key pair from a virtual server by using the unbind ssl certKey <`certkeyName`> command, an error message appears because the syntax of the command has changed. At the command prompt, type the following command:

```
unbind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

```
unbind ssl vserver vssl -certkeyName sslckey
<!--NeedCopy-->
```

Bind an SSL certificate-key pair to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open an SSL virtual server and, in **Advanced Settings**, click **SSL Certificate**.
3. Bind a server certificate or CA certificate to the virtual server. To add a server certificate as an SNI certificate, select Server Certificate for SNI.

SSL virtual server parameters

You can set the advanced SSL configuration for an SSL virtual server. Many of these parameters can also be set in an SSL profile. For information about the parameters that can be set in an SSL profile, see SSL profile parameters.

Set SSL virtual server parameters by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName>@ [-clearTextPort <port>] [-dh ( ENABLED | DISABLED ) -dhFile <string>] [-dhCount <positive_integer>][-dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED | DISABLED) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED | DISABLED )[-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED | DISABLED ) [-cipherURL <URL>]] [-sslv2Redirect ( ENABLED | DISABLED )[-sslv2URL <URL>]] [-clientAuth ( ENABLED | DISABLED ) [-clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED | DISABLED )][-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl2 ( ENABLED | DISABLED )] [-ssl3 ( ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 ( ENABLED | DISABLED )] [-tls12 ( ENABLED | DISABLED )][-tls13 ( ENABLED | DISABLED )] [-SNIEnable ( ENABLED | DISABLED )][-ocspStapling ( ENABLED | DISABLED )] [-pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES | NO )] [-dtlsProfileName <string>] [-sslProfile <string>] [-HSTS ( ENABLED | DISABLED )][-maxage <positive_integer>] [-IncludeSubdomains ( YES |
NO )][-strictSigDigestCheck ( ENABLED | DISABLED )] [-zeroRttEarlyData (ENABLED | DISABLED )] [-tls13SessionTicketsPerAuthContext <positive_integer>] [-dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

Diffe-Hellman (DH) parameters

If you are using ciphers on the appliance that require a DH key exchange to set up the SSL transaction, enable DH key exchange on the appliance and configure other settings based on your network.

To list the ciphers for which DH parameters must be set by using the CLI, type: sh cipher DH.

To list the ciphers for which DH parameters must be set by using the configuration utility, navigate to **Traffic Management > SSL > Cipher Groups**, and double-click **DH**.

For details on how to enable DH key exchange, see Generate a Diffie-Hellman (DH) key.

Configure DH parameters by using the CLI

At the command prompt, type the following commands to configure DH parameters and verify the configuration:

```
set ssl vserver <vserverName> -dh <Option> -dhCount <RefreshCountValue> -filepath <string>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-server -dh ENABLED  -dhFile /nsconfig/ssl/ns-server.cert -dhCount 1000

Done

show ssl vserver vs-server

            Advanced SSL configuration for VServer vs-server:

            DH: ENABLED

            Ephemeral RSA: ENABLED          Refresh Count: 1000

            Session Reuse: ENABLED          Timeout: 120 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            ClearText Port: 0

            Client Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            OCSP Stapling: DISABLED

            HSTS: DISABLED

            HSTS IncludeSubDomains: NO

            HSTS Max-Age: 0

            SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED  TLSv1.2: ENABLED

    1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Configure DH Parameters by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable DH Param**, and specify a refresh count and file path.

Ephemeral RSA

Ephemeral RSA allows export clients to communicate with the secure server even if the server certificate does not support export clients (1024-bit certificate). If you want to prevent export clients from accessing the secure web object and/or resource, you need to disable ephemeral RSA key exchange.

By default, this feature is enabled on the NetScaler appliance, with the refresh count set to zero (infinite use).

> **Note:**
>
> The ephemeral RSA key is automatically generated when you bind an export cipher to an SSL or TCP-based SSL virtual server or service. When you remove the export cipher, the eRSA key is not deleted but reused later when another export cipher is bound to an SSL or TCP-based SSL virtual server or service. The eRSA key is deleted when the system restarts.

Configure Ephemeral RSA by using the CLI

At the command prompt, type the following commands to configure ephemeral RSA and verify the configuration:

```
set ssl vserver <vServerName> -eRSA (enabled | disabled) -eRSACount <positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-server -eRSA ENABLED -eRSACount 1000
Done
show ssl vserver vs-server

        Advanced SSL configuration for VServer vs-server:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED  TLSv1.2: ENABLED

1)      Cipher Name: DEFAULT
```

```
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

Configure Ephemeral RSA by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable Ephemeral RSA**, and specify a refresh count.

Session reuse

For SSL transactions, establishing the initial SSL handshake requires CPU-intensive public key encryption operations. Most handshake operations are associated with the exchange of the SSL session key (client key exchange message). When a client session is idle for some time and is then resumed, the SSL handshake is typically conducted all over again. With session reuse enabled, session key exchange is avoided for session resumption requests received from the client.

Session reuse is enabled on the NetScaler appliance by default. Enabling this feature reduces server load, improves response time, and increases the number of SSL transactions per second (TPS) that can be supported by the server.

Configure session reuse by using the CLI

At the command prompt, type the following commands to configure session reuse and verify the configuration:

```
set ssl vserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -
    sessTimeout <positive_integer>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -sessreuse enabled -sesstimeout 600
Done

show ssl vserver vs-ssl

        Advanced SSL configuration for VServer vs-ssl:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 1000
        Session Reuse: ENABLED          Timeout: 600 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
        SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED  TLSv1.2: ENABLED

1)      CertKey Name: Auth-Cert-1       Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

Configure session reuse by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable Session Reuse**, and specify a time for which to keep the session active.

SSL protocol settings

The NetScaler appliance supports the SSLv3, TLSv1, TLSv1.1, and TLSv1.2 protocols. Each of these can be set on the appliance as required by your deployment and the type of clients that connect to the appliance.

TLS protocol versions 1.0, 1.1, and 1.2 are more secure than older versions of the TLS/SSL protocol. However, to support legacy systems, many TLS implementations maintain backward compatibility with the SSLv3 protocol. In an SSL handshake, the highest protocol version common to the client and the SSL virtual server configured on the NetScaler appliance is used.

In the first handshake attempt, a TLS client offers the highest protocol version that it supports. If the handshake fails, the client offers a lower protocol version. For example, if a handshake with TLS version 1.1 is not successful, the client attempts to renegotiate by offering the TLSv1.0 protocol. If that attempt is unsuccessful, the client reattempts with the SSLv3 protocol. A “man in the middle” (MITM) attacker can break the initial handshake and trigger renegotiation with the SSLv3 protocol, and then exploit a vulnerability in SSLv3. To mitigate such attacks, you can disable SSLv3 or not allow renegotiation using a downgraded protocol. However, this might not be practical if your deployment includes legacy systems. An alternative is to recognize a signaling cipher suite value (TLS_FALLBACK_SCSV) in the client request.

A TLS_FALLBACK_SCSV value in a client hello message indicates to the virtual server that the client has previously attempted to connect with a higher protocol version and that the current request is a fallback. If the virtual server detects this value, and it supports a version higher than the one indicated by the client, it rejects the connection with a fatal alert. If a TLS_FALLBACK_SCSV is not included in the client hello message, or if the protocol version in the client hello is the highest protocol version supported by the virtual server, the handshake succeeds.

Configure SSL protocol support by using the CLI

At the command prompt, type the following commands to configure SSL protocol support and verify the configuration:

```
set ssl vserver <vServerName> -ssl2 ( ENABLED | DISABLED ) -ssl3 (
    ENABLED | DISABLED ) -tls1 ( ENABLED | DISABLED ) -tls11 ( ENABLED |
     DISABLED ) -tls12 ( ENABLED | DISABLED )

show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -tls11 ENABLED -tls12 ENABLED

Done

sh ssl vs vs-ssl

        Advanced SSL configuration for VServer vs-ssl:

            DH: DISABLED

            Ephemeral RSA: ENABLED                            Refresh
       Count: 0

            Session Reuse: ENABLED                              Timeout
       : 120 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            ClearText Port: 0

            Client Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            SSLv2: DISABLED        SSLv3: ENABLED    TLSv1.0: ENABLED    TLSv1.1: ENABLED  TLSv1.2: ENABLED

            Push Encryption Trigger: Always

            Send Close-Notify: YES

            1 bound certificate:

    1)      CertKey Name: mycert  Server Certificate

            1 configured cipher:

    1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Configure SSL protocol support by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select a protocol to enable.

Close-notify

A close-notify is a secure message that indicates the end of SSL data transmission. A close-notify setting is required at the global level. This setting applies to all virtual servers, services, and service groups. For information about the global setting, see the **Global SSL parameters** section later in this page.

In addition to the global setting, you can set the close-notify parameter at the virtual server, service, or service group level. You therefore have the flexibility of setting the parameter for one entity and unsetting it for another entity. However, make sure that you set this parameter at the global level. Otherwise, the setting at the entity level does not apply.

Configure the close-notify parameter at the entity level by using the CLI

At the command prompt, type any of the following commands to configure the close-notify parameter and verify the configuration:

1. To configure the close-notify parameter at the virtual server level, type:

```
set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )
show ssl vserver <vServerName>
<!--NeedCopy-->
```

1. To configure the close-notify parameter at the service level, type:

```
set ssl service <serviceName> -sendCloseNotify ( YES | NO )
show ssl service <serviceName>
<!--NeedCopy-->
```

1. To configure the close-notify parameter at the service group level, type:

```
set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver sslvsvr -sendCloseNotify YES

Done
<!--NeedCopy-->
```

Configure the close-notify parameter at the entity level by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Send Close-Notify**.

Global SSL parameters

Advanced customization of your SSL configuration addresses specific issues. You can use the `set ssl parameter` command or the configuration utility to specify the following:

- Quantum size to be used for SSL transactions.
- CRL memory size.
- OCSP cache size.
- Deny SSL renegotiation.
- Set the PUSH flag for decrypted, encrypted, or all records.
- Drop requests if the client initiates the handshake for one domain and sends an HTTP request for another domain.
- Set the time after which encryption is triggered.
  Note: The time that you specify applies only if you use the
  `set ssl vserver` command or the configuration utility to set timer-based encryption.

Configure global SSL parameters by using the CLI

At the command prompt, type the following commands to configure advanced SSL settings and verify the configuration:

```
set ssl parameter [-quantumSize <quantumSize>] [-crlMemorySizeMB <
    positive_integer>] [-strictCAChecks (YES | NO)] [-sslTriggerTimeout
    <positive_integer>] [-sendCloseNotify (YES | NO)] [-
    encryptTriggerPktCount <positive_integer>] [-denySSLReneg <
    denySSLReneg>] [-insertionEncoding (Unicode|UTF-8)] [-ocspCacheSize
    <positive_integer>][- pushFlag <positive_integer>] [-
    dropReqWithNoHostHeader (YES | NO)] [-pushEncTriggerTimeout <
    positive_integer>]
show ssl parameter
<!--NeedCopy-->
```

Example:

```
set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCAChecks
    no -sslt
iggerTimeout 100 -sendClosenotify no -encryptTriggerPktCount 45 -
    denySSLReneg no
-insertionEncoding unicode -ocspCacheSize 10 -pushFlag 3 -
    dropReqWithNoHostHeader YES  -pushEncTriggerTimeout 100 ms
Done

show ssl parameter
Advanced SSL Parameters
-----------------------
        SSL quantum size:               8 kB
        Max CRL memory size:            256 MB
        Strict CA checks:               NO
        Encryption trigger timeout      100 mS
        Send Close-Notify               NO
        Encryption trigger packet count:        45
        Deny SSL Renegotiation          NO
        Subject/Issuer Name Insertion Format:    Unicode
        OCSP cache size:        10 MB
           Push flag:      0x3 (On every decrypted and encrypted record )
                                Strict Host Header check for SNI enabled SSL sessions:          YES
                                 PUSH encryption trigger timeout 100 ms
 Done
<!--NeedCopy-->
```

Configure global SSL parameters by using the GUI

Navigate to **Traffic Management** > **SSL** and, in the **Settings** group, select **Change advanced SSL settings**.

PUSH flag based encryption trigger mechanism

The encryption trigger mechanism that is based on the PSH TCP flag now enables you to do the following:

- Merge consecutive packets in which the PSH flag is set into a single SSL record, or ignore the PSH flag.
- Perform timer-based encryption, in which the time-out value is set globally by using the set ssl parameter -pushEncTriggerTimeout command.

Configure PUSH flag-based encryption by using the CLI

At the command prompt, type the following commands to configure PUSH flag-based encryption and verify the configuration:

```
set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]

show ssl vserver
<!--NeedCopy-->
```

Example:

```
set ssl vserver vserver1 -pushEncTrigger always

Done
```

```

sh ssl vserver vserver1

            Advanced SSL configuration for VServer vserver1:

            DH: DISABLED

            DH Private-Key Exponent Size Limit: DISABLED    Ephemeral RSA: ENABLED      Refresh Count: 0

            Session Reuse: ENABLED          Timeout: 120 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            ClearText Port: 0

            Client Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            OCSP Stapling: DISABLED

            HSTS: DISABLED

            HSTS IncludeSubDomains: NO

            HSTS Max-Age: 0

            SSLv2: DISABLED  SSLv3: ENABLED  TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2: ENABLED  TLSv1.3: DISABLED

            Push Encryption Trigger: Always

            Send Close-Notify: YES

            Strict Sig-Digest Check: DISABLED

            Zero RTT Early Data: DISABLED

            DHE Key Exchange With PSK: NO

            Tickets Per Authentication Context: 1

            ECC Curve: P_256, P_384, P_224, P_521

    1)      Cipher Name: DEFAULT

            Description: Default cipher list with encryption strength >= 128bit

Done
<!--NeedCopy-->
```

Configure PUSH flag-based encryption by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual servers** and open an SSL virtual server.
2. In the **SSL Parameters** section, from the **PUSH Encryption Trigger** list, select a value.

Support for TLS1.2 signature hash algorithm

The NetScaler appliance is completely TLS1.2 signature hash extension compliant.

In an SSL handshake, a client sends a list of supported signature hash (sighash) algorithms. The client uses the "signature_algorithms" extension to indicate to the server which sighash algorithm pairs may be used in the SSL handshake messages (SKE and CCV). The "extension_data" field of this extension contains a "supported_signature_algorithms" value in the Client Hello message. The SSL handshake proceeds if the server supports one of these sighash algorithms. If the server does not support any of these algorithms, the connection is dropped.

Similarly, if a client certificate is requested by the server for client authentication, the Certificate Request message contains a "supported_signature_algorithms" value. The client certificate is selected based on this sighash algorithm.

> **Note:**
>
> The NetScaler appliance acts as a server to a client and as a client to the back-end server.

Previously. the appliance supported only RSA-SHA1 and RSA-SHA256 on the front end, and RSA-MD5, RSA-SHA1, and RSA-SHA256 on the back end. In addition, the VPX appliance supported DSA-SHA1 on the front end and back end.

Now, the MPX/SDX/VPX appliance supports the following sighash combinations. On an SDX appliance, if an SSL chip is assigned to a VPX instance, the cipher support of an MPX appliance applies. Otherwise, the normal cipher support of a VPX instance applies.

- On a VPX instance:
    - RSA-MD5
    - RSA-SHA1
    - RSA-SHA224
    - RSA-SHA256
    - RSA-SHA384
    - RSA-SHA512
    - DSA-SHA1
    - DSA-SHA224
    - DSA-SHA256
    - DSA-SHA384
    - DSA-SHA512
- On an MPX/SDX appliance with N3 chips:
    - RSA-MD5
    - RSA-SHA1
    - RSA-SHA224
    - RSA-SHA256
    - RSA-SHA384
    - RSA-SHA512
    - ECDSA-SHA1
    - ECDSA-SHA224
    - ECDSA-SHA256
    - ECDSA-SHA384
    - ECDSA-SHA512
- On an MPX/SDX appliance without N3 chips:
    - RSA-MD5
    - RSA-SHA1
    - RSA-SHA224
    - RSA-SHA256
    - RSA-SHA384
    - RSA-SHA512

By default, all the sighash algorithms are enabled. However, you can enable only a few sighash algorithms by using the following command:

```
set ssl parameter -sigDigestType <sigDigestType>

Parameters
```

```

sigDigestType

Signature digest algorithms supported by the appliance. The platform determines the list of algorithms supported by default.

                On VPX: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-

                SHA512 DSA-SHA1 DSA-SHA224 DSA-SHA256 DSA-SHA384 DSA-SHA512

                On MPX with N3 cards: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-

                SHA256 RSA-SHA384 RSA-SHA512 ECDSA-SHA1 ECDSA-SHA224 ECDSA-

                SHA256 ECDSA-SHA384 ECDSA-SHA512

                Other MPX Platforms: RSA-MD5 RSA-SHA1 RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-

                SHA512.

    set ssl parameter -sigDigestType RSA-SHA224 RSA-SHA256 RSA-SHA384 RSA-SHA512
<!--NeedCopy-->
```

Validate the peer certificate

According to RFC 5246, the peer certificate must be signed using one of the sighash algorithms included in the Client Hello extension. To do this, use the strictSigDigestCheck parameter. Depending on the sighash list sent by the client, if you enable strictSigDigestCheck, the appliance returns a certificate signed by one of the sighash algorithms mentioned in the Client Hello extension. If the peer does not have a proper certificate, the connection is dropped. If this parameter is disabled, sighash is not checked in the peer certificate.

You can configure strict signature digest check on an SSL virtual server and service. If you enable this parameter on an SSL virtual server, the server certificate sent by the server must be signed by one of the sighash algorithms listed in the Client Hello extension. If client authentication is enabled, then the client certificate received by the server must be signed using one of the sighash algorithms listed in the certificate request sent by the server.

If you enable this parameter on an SSL service, the server certificate received by the client must be signed by one of the sighash algorithms listed in the Client Hello extension. The client certificate must be signed using one of the sighash algorithms listed in the certificate request message.

If default profile is enabled, you can use it to configure strict signature digest check on an SSL virtual server, SSL service, and SSL profile.

Configure strict signature digest check on an SSL virtual server, service, or profile by using the CLI

At the command prompt, type:

```
set ssl vserver <vServerName> -strictSigDigestCheck ( ENABLED | DISABLED )

set ssl service <serviceName> -strictSigDigestCheck ( ENABLED | DISABLED )

set ssl profile <name>-strictSigDigestCheck ( ENABLED | DISABLED )

Parameters

strictSigDigestCheck

              Check whether peer entity certificate is signed using one of the signature-hash algorithms supported by the NetScaler appliance.

              Possible values: ENABLED, DISABLED

              Default: DISABLED
<!--NeedCopy-->
```

Example:

```
set ssl vserver v1  – strictSigDigestCheck Enabled
set ssl service s1  – strictSigDigestCheck Enabled
set ssl profile p1  – strictSigDigestCheck Enabled
<!--NeedCopy-->
```

> **Important:**
>
> If DH, ECDHE, or ECDSA ciphers are configured on the appliance, the SKE message must be signed using one of the sighash common to the client list and the list configured on the appliance. If there is no common sighash, the connection is dropped.

How-to articles

September 21, 2020

How-to articles are simple and easy to use articles with configuration steps for common deployments. Click a link below to view the article.

Configure per directory client authentication

Configure support for Outlook web access

Configure SSL based header insertion

Configure SSL offloading with end-to-end encryption

Configure transparent SSL acceleration

Configure SSL acceleration with HTTP on the front end and SSL on the back end

Configure SSL offloading with other TCP protocols

Configure SSL bridging

Configure SSL monitoring when client authentication is enabled on the backend service

Configure a secure content switching server

Configure an HTTPS virtual server to accept HTTP traffic

Configure graceful cleanup of SSL sessions

Configure SSLv2 redirection

Configure synchronization of files in a high availability setup

SSL certificates

September 21, 2020

An SSL certificate, which is an integral part of any SSL transaction, is a digital data form (X509) that identifies a company (domain) or an individual. The certificate has a public key component that is visible to any client that wants to initiate a secure transaction with the server. The corresponding private key, which resides securely on the NetScaler appliance, is used to complete asymmetric key (or public key) encryption and decryption.

You can obtain an SSL certificate and key in either of the following ways:

- From an authorized certificate authority (CA), such as Verisign
- By generating a new SSL certificate and key on the NetScaler appliance

Alternately, you can use an existing SSL certificate on the appliance.

Cautions:

- Citrix recommends that you use certificates obtained from authorized CAs, such as Verisign, for all your SSL transactions. Use certificates generated on the NetScaler appliance for testing purposes only, not in any live deployment.
- If while adding a certificate-key pair, you add a certificate file with the same name as an existing certificate file but with different content, the original certificate file is overwritten with no warning. This might cause issues after the appliance is restarted because the original certificate file is no longer available in the /nsconfig/ssl directory.

Create a certificate

September 21, 2020

A certificate authority (CA) is an entity that issues digital certificates for use in public key cryptography. Certificates issued or signed by a CA are automatically trusted by applications, such as web browsers, that conduct SSL transactions. These applications maintain a list of the CAs that they trust. If the certificate being used for the secure transaction is signed by any of the trusted CAs, the application proceeds with the transaction.

To obtain an SSL certificate from an authorized CA, create a private key. Use that key to create a certificate signing request (CSR), and submit the CSR to the CA. The only special characters allowed in the file names are underscore and dot.

To import an existing certificate and key, see Import a Certificate.

Create a private key

The private key is the most important part of a digital certificate. By definition, this key is not to be shared with anyone and must be kept securely on the NetScaler appliance. Any data encrypted with the public key can be decrypted only by using the private key.

The appliance supports two encryption algorithms, RSA and DSA, for creating private keys. You can submit either type of private key to the CA. The certificate that you receive from the CA is valid only with the private key that was used to create the CSR. The key is required for adding the certificate to the NetScaler appliance.

Caution: Be sure to limit access to your private key. Anyone who has access to your private key can decrypt your SSL data.

All SSL certificates and keys are stored in the `/nsconfig/ssl` folder on the appliance. For added security, you can use the Data Encryption Standard (DES) or triple DES (3DES) algorithm to encrypt the private key stored on the appliance.

> **Note:**
>
> The length of the SSL key name allowed includes the length of the absolute path name if the path is included in the key name.

Create an RSA private key by using the CLI

At the command prompt, type:

```
create ssl rsakey <keyFile> <bits> [-exponent ( 3 | F4 )] [-keyform ( DER | PEM )]
<!--NeedCopy-->
```

Example:

```
create rsakey testkey 2048 -exponent F4 -keyform PEM
<!--NeedCopy-->
```

Create an RSA private key by using the GUI

Navigate to **Traffic Management > SSL** and, in the **SSL Keys** group, select **Create RSA Key**, and create an RSA key.

Create a DSA private key by using the CLI

At the command prompt, type:

```
create ssl dsakey <keyfile> <bits> [-keyform (DER | PEM)]
<!--NeedCopy-->
```

Example:

```
create ssl dsakey Key-DSA-1 2048 -keyform PEM
<!--NeedCopy-->
```

Create a DSA private key by using the GUI

Navigate to **Traffic Management > SSL** and, in the **SSL Keys** group, select **Create DSA Key**, and create a DSA key.

Support for SHA 256 digest algorithm in a certificate signing request

The certificate signing request (CSR) is a collection of information. It includes the domain name, other important company details, and the public key to be used to create the certificate. To avoid generating an invalid certificate, make sure that the details you provide are accurate.

The NetScaler appliance supports creating a CSR signed with the SHA1 digest algorithm by default. In earlier releases, to create a CSR signed with the SHA256 digest algorithm, you had to use OpenSSL.

The appliance supports creating a CSR signed with the SHA256 digest algorithm. The encryption hash algorithm used in SHA256 makes it stronger than SHA1.

Create a certificate signing request by using the CLI

At the command prompt, type:

```
create ssl certreq <reqFile> -keyFile <input_filename> | -fipsKeyName <
    string>) [-keyForm (DER | PEM) {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName <string>
     -organizationUnitName <string> -localityName <string> -commonName
     <string> -emailAddress <string> {
 -challengePassword  }
  -companyName <string> -digestMethod ( SHA1 | SHA256 )
<!--NeedCopy-->
```

Example:

```
create ssl certreq priv_csr_sha256 -keyfile priv_2048_2 -keyform PEM -
    countryName IN -stateName Karnataka -localityName Bangalore -
    organizationName Citrix -organizationUnitName NS -digestMethod
    SHA256
<!--NeedCopy-->
```

Create a certificate signing request by using the GUI

1. Navigate to **Traffic Management > SSL**.
2. In **SSL** Certificate, click **Create Certificate Signing Request (CSR)**.
3. Specify values for the different parameters.
4. In **Digest** Method, select **SHA256**.

Submitting the CSR to the CA

Most CAs accept certificate submissions by email. The CA returns a valid certificate to the email address from which you submit the CSR.

Generate a test certificate

> **Note:**
>
> To generate a server test certificate, see Generating a Server Test Certificate.

The NetScaler appliance has a built-in CA tools suite that you can use to create self-signed certificates for testing purposes.

Caution: Because these certificates are signed by the NetScaler appliance itself, not by an actual CA, you must not use them in a production environment. If you attempt to use a self-signed certificate in a production environment, users receive a "certificate invalid" warning each time the virtual server is accessed.

The appliance supports creation of the following types of certificates

- Root-CA certificates
- Intermediate-CA certificates
- End-user certificates
  - server certificates
  - client certificates

Before generating a certificate, create a private key and use that to create a certificate signing request (CSR) on the appliance. Then, instead of sending the CSR out to a CA, use the NetScaler CA Tools to generate a certificate.

Create a certificate by using a wizard

1. Navigate to **Traffic Management > SSL**.
2. In the details pane, under **Getting Started**, select the wizard for the type of certificate that you want to create.
3. Follow the instructions on the screen.

Create a root-CA certificate by using the CLI

At the command prompt, type the following command:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
<!--NeedCopy-->
```

In the following example, csreq1 is the CSR and rsa1 is the private key that was created earlier.

Example:

```
create ssl cert cert1 csreq1 ROOT_CERT -keyFile rsa1 -keyForm PEM -days
    365

  Done
<!--NeedCopy-->
```

Create an intermediate-CA certificate by using the CLI

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <
    input_filename>] [-keyform ( DER | PEM )] [-days <positive_integer>]
     [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm (
     DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )]
    [-CAserial <output_filename>]
<!--NeedCopy-->
```

In the following example, csr1 is the CSR created earlier. Cert1 and rsakey1 are the certificate and corresponding key of the self-signed (root-CA) certificate, and pvtkey1 is the private key of the intermediate-CA certificate.

Example:

```
create ssl cert certsy csr1 INTM_CERT -CAcert cert1 -CAkey rsakey1 -
    CAserial 23


Done

create ssl rsakey pvtkey1 2048 -exponent F4 -keyform PEM
<!--NeedCopy-->
```

Create a root-CA certificate by using the GUI

Navigate to **Traffic Management > SSL** and, in the **Getting Started** group, select **Root-CA Certificate Wizard**, and configure a root CA certificate.

Create an intermediate-CA certificate by using the GUI

Navigate to **Traffic Management > SSL** and, in the **Getting Started** group, select **Intermediate-CA Certificate Wizard**, and configure an intermediate CA certificate.

Create an end-user certificate

An end-user certificate can be a client certificate or a server certificate. To create a test end-user certificate, specify the Intermediate CA certificate or the self-signed root-CA certificate.

Note: To create an end-user certificate for production use, specify a trusted CA certificate and send the CSR to a certificate authority (CA).

Create a test end-user certificate by using the CLI

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform ( DER | PEM )] [-days<positive_integer>] [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey<input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <output_filename>]
<!--NeedCopy-->
```

If there is no intermediate certificate, use the certificate (cert1) and private key (rsakey1) values of the root-CA certificate in CAcert and CAkey.

Example:

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert cert1 -CAkey rsakey1 -CAserial 23

Done
<!--NeedCopy-->
```

If there is an intermediate certificate, use the certificate (certsy) and private key (pvtkey1) values of the intermediate certificate in CAcert and CAkey.

Example:

```
create ssl cert cert12 csr1 SRVR_CERT -CAcert certsy -CAkey pvtkey1 -CAserial 23

Done
<!--NeedCopy-->
```

Install, link, and update certificates

September 21, 2020

To install a certificate, see Add or update a certificate-key pair.

Link certificates

If an intermediate CA that is not recognized by standard web browsers as a trusted CA, issues the server certificate, the CA certificate(s) must be sent to the client with the server's own certificate. Otherwise, the browser terminates the SSL session because it fails to authenticate the server certificate.

There are two ways to add the server and intermediate certificates:

- Create a certificate set that contains the chain of certificates.
- Create a chain of certificates manually by adding and linking the certificates individually.

Add and link a certificate set

Note: This feature is not supported on the NetScaler FIPS platform and in a cluster setup.

Instead of adding and linking individual certificates, you can now group a server certificate and up to nine intermediate certificates in a single file. You can specify the file's name when adding a certificate-key pair. Before you do so, make sure that the following prerequisites are met.

- The certificates in the file are in the following order:
  - Server certificate (should be the first certificate in the file)
  - Optionally, a server key
  - Intermediate certificate 1 (ic1)
  - Intermediate certificate 2 (ic2)
  - Intermediate certificate 3 (ic3), and so on
    Note: Intermediate certificate files are created for each intermediate certificate with the name "<certificatebundlename>.pem_ic<
    n>" where n is between 1 and 9. For example, bundle.pem_ic1, where bundle is the name of the certificate set and ic1 is the first intermediate certificate in the set.
- Bundle option is selected.
- No more than nine intermediate certificates are present in the file.

The file is parsed and the server certificate, intermediate certificates, and server key (if present) are identified. First, the server certificate and key are added. Then, the intermediate certificates are added, in the order in which they were added to the file, and linked accordingly.

An error is reported if any of the following conditions exist:

- A certificate file for one of the intermediate certificates exists on the appliance.
- The key is placed before the server certificate in the file.
- An intermediate certificate is placed before the server certificate.
- Intermediate certificates are not in placed in the file in the same order as they are created.
- No certificates are present in the file.
- A certificate is not in the proper PEM format.
- The number of intermediate certificates in the file exceeds nine.

Add a certificate set by using the CLI

At the command prompt, type the following commands to create a certificate set and verify the configuration:

```
add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES
     | NO)

show ssl certKey

show ssl certlink
<!--NeedCopy-->
```

In the following example, the certificate set (bundle.pem) contains the following files:

Server certificate (bundle) linked to bundle_ic1

First intermediate certificate (bundle_ic1) linked to bundle_ic2

Second intermediate certificate (bundle_ic2) linked to bundle_ic3

Third intermediate certificate (bundle_ic3)

```
add ssl certKey bundle -cert bundle.pem -key bundle.pem -bundle yes

sh ssl certkey

1)      Name: ns-server-certificate

        Cert Path: ns-server.cert

        Key Path: ns-server.key

        Format: PEM

        Status: Valid,   Days to expiration:5733

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Server Certificate

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=default OULLFT

        Validity

                Not Before: Apr 21 15:56:16 2016 GMT

                Not After : Mar  3 06:30:56 2032 GMT

        Subject:  C=US,ST=California,L=San Jose,O=Citrix ANG,OU=NS Internal,CN=default OULLFT

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

2)      Name: servercert

        Cert Path: complete/server/server_rsa_1024.pem

        Key Path: complete/server/server_rsa_1024.ky

        Format: PEM

        Status: Valid,   Days to expiration:7150

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Server Certificate

        Version: 3

        Serial Number: 1F

        Signature Algorithm: sha1WithRSAEncryption

        Issuer:  C=IN,ST=KAR,O=Citrix R&D Pvt Ltd,CN=Citrix

        Validity

                Not Before: Sep  2 09:54:07 2008 GMT

                Not After : Jan 19 09:54:07 2036 GMT

        Subject:  C=IN,ST=KAR,O=Citrix Pvt Ltd,CN=Citrix

        Public Key Algorithm: rsaEncryption

        Public Key size: 1024

3)      Name: bundletest

        Cert Path: bundle9.pem

        Key Path: bundle9.pem

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Server Certificate

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA9

        Validity

                Not Before: Nov 28 06:43:11 2014 GMT

                Not After : Nov 25 06:43:11 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=Server9

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

```

```
4)      Name: bundletest_ic1

        Cert Path: bundle9.pem_ic1

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA8

        Validity

                Not Before: Nov 28 06:42:56 2014 GMT

                Not After : Nov 25 06:42:56 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA9

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

5)      Name: bundletest_ic2

        Cert Path: bundle9.pem_ic2

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA7

        Validity

                Not Before: Nov 28 06:42:55 2014 GMT

                Not After : Nov 25 06:42:55 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA8

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

6)      Name: bundletest_ic3

        Cert Path: bundle9.pem_ic3

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA6

        Validity

                Not Before: Nov 28 06:42:53 2014 GMT

                Not After : Nov 25 06:42:53 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA7

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

7)      Name: bundletest_ic4

        Cert Path: bundle9.pem_ic4

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA5

        Validity

                Not Before: Nov 28 06:42:51 2014 GMT

                Not After : Nov 25 06:42:51 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA6

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048
```

```

8)      Name: bundletest_ic5

        Cert Path: bundle9.pem_ic5

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA4

        Validity

                Not Before: Nov 28 06:42:50 2014 GMT

                Not After : Nov 25 06:42:50 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA5

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

9)      Name: bundletest_ic6

        Cert Path: bundle9.pem_ic6

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA3

        Validity

                Not Before: Nov 28 06:42:48 2014 GMT

                Not After : Nov 25 06:42:48 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA4

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

10)     Name: bundletest_ic7

        Cert Path: bundle9.pem_ic7

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA2

        Validity

                Not Before: Nov 28 06:42:46 2014 GMT

                Not After : Nov 25 06:42:46 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA3

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

11)     Name: bundletest_ic8

        Cert Path: bundle9.pem_ic8

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=ICA1

        Validity

                Not Before: Nov 28 06:42:45 2014 GMT

                Not After : Nov 25 06:42:45 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA2

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

12)     Name: bundletest_ic9

        Cert Path: bundle9.pem_ic9

        Format: PEM

        Status: Valid,   Days to expiration:3078

        Certificate Expiry Monitor: ENABLED

        Expiry Notification period: 30 days

        Certificate Type: Intermediate CA

        Version: 3

        Serial Number: 01

        Signature Algorithm: sha256WithRSAEncryption

        Issuer:  C=IN,ST=ka,O=sslteam,CN=RootCA4096

        Validity

                Not Before: Nov 28 06:42:43 2014 GMT

                Not After : Nov 25 06:42:43 2024 GMT

        Subject:  C=IN,ST=ka,O=sslteam,CN=ICA1

        Public Key Algorithm: rsaEncryption

        Public Key size: 2048

Done

sh ssl certlink

1)      Cert Name: bundletest    CA Cert Name: bundletest_ic1

2)      Cert Name: bundletest_ic1        CA Cert Name: bundletest_ic2

3)      Cert Name: bundletest_ic2        CA Cert Name: bundletest_ic3
```

```

4)      Cert Name: bundletest_ic3       CA Cert Name: bundletest_ic4

5)      Cert Name: bundletest_ic4       CA Cert Name: bundletest_ic5

6)      Cert Name: bundletest_ic5       CA Cert Name: bundletest_ic6

7)      Cert Name: bundletest_ic6       CA Cert Name: bundletest_ic7

8)      Cert Name: bundletest_ic7       CA Cert Name: bundletest_ic8

9)      Cert Name: bundletest_ic8       CA Cert Name: bundletest_ic9

Done
<!--NeedCopy-->
```

Add a certificate set by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Certificates**.
2. In the **SSL Certificates** pane, click **Install**.
3. In the **Install Certificate** dialog box, type the details, such as the certificate and key file name, and then select **Certificate Bundle**.
4. Click **Install**, and then click **Close**.

Create a chain of certificates

Instead of using a set of certificates (a single file), you can create a chain of certificates. The chain links the server certificate to its issuer (the intermediate CA). For this approach to work, the intermediate CA certificate file must already be installed on the NetScaler appliance, and the client application must trust one of the certificates in the chain. For example, link Cert-Intermediate-A to Cert-Intermediate-B, where Cert-Intermediate-B is linked to Cert-Intermediate-C, which is a certificate trusted by the client application.

Note: The appliance supports sending a maximum of 10 certificates in the chain of certificates sent to the client (one server certificate and nine CA certificates).

Create a certificate chain by using the CLI

At the command prompt, type the following commands to create a certificate chain and verify the configuration. (Repeat the first command for each new link in the chain.)

```
link ssl certkey <certKeyName> <linkCertKeyName>
```

```
show ssl certlink
<!--NeedCopy-->
```

Example:

```
link ssl certkey siteAcertkey CAcertkey
 Done

show ssl certlink

linked certificate:
       1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
Done
<!--NeedCopy-->
```

Create a certificate chain by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Certificates**.
2. Select a server certificate, and in the **Action** list, select **Link**, and specify a CA certificate name.

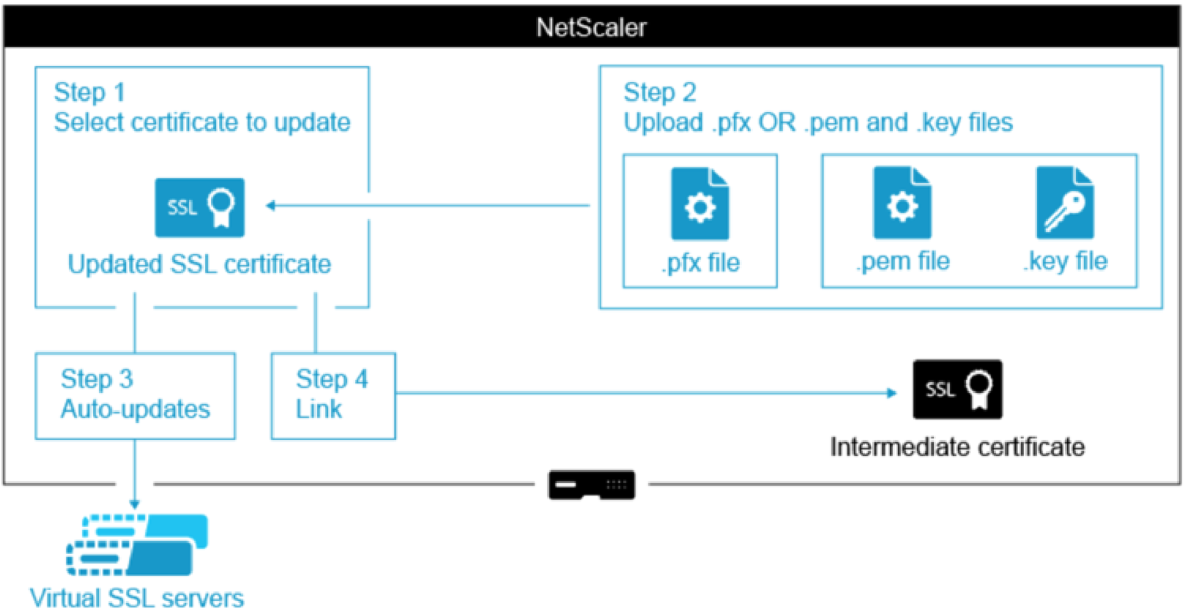
Update an existing server certificate

To avoid downtime when replacing a certificate-key pair, you can update an existing certificate. If you want to replace a certificate with a certificate that was issued to a different domain, you must disable domain checks before updating the certificate.

To receive notifications about certificates due to expire, you can enable the expiry monitor.

When you remove or unbind a certificate from a configured SSL virtual server, or an SSL service, the virtual server or service becomes inactive until a new valid certificate is bound to it. To avoid downtime, you can use the update feature to replace a certificate-key pair that is bound to an SSL virtual server or an SSL service, without first unbinding the existing certificate.

Overview diagram of how to update an SSL certificate on the NetScaler appliance.


NetScaler
Step 1
Select certificate to update
SSL
Updated SSL certificate
Step 2
Upload .pfx OR .pem and .key files
.pfx file
.pem file
.key file
Step 3
Auto-updates
Step 4
Link
SSL
Intermediate certificate
Virtual SSL servers

Update an existing certificate-key pair by using the CLI

At the command prompt, type the following commands to update an existing certificate-key pair and verify the configuration:

```
update ssl certkey <certkeyName> -cert <string> -key <string>

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
    nsconfig/ssl/pkey.pem

Done

show ssl certkey siteAcertkey

Name: siteAcertkey       Status: Valid

           Version: 3

           Serial Number: 02

           Signature Algorithm: md5WithRSAEncryption

           Issuer: /C=US/ST=CA/L=Santa Clara/O=siteA/OU=Tech

           Validity

                Not Before: Nov 11 14:58:18 2001 GMT

                Not After: Aug 7 14:58:18 2004 GMT

           Subject: /C=US/ST-CA/L=San Jose/O=CA/OU=Security

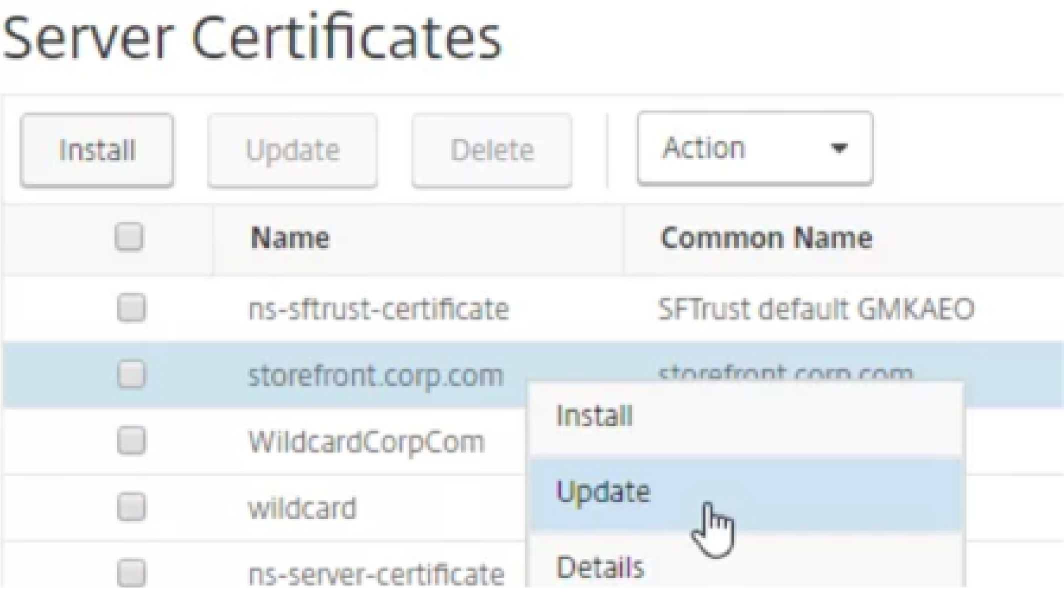
           Public Key Algorithm: rsaEncryption

           Public Key size: 2048

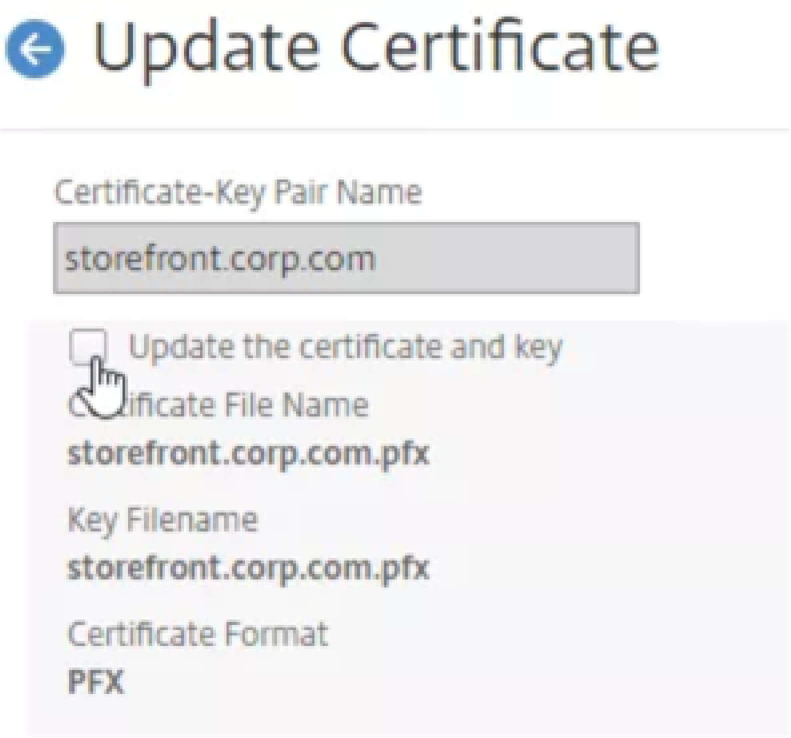
Done
<!--NeedCopy-->
```

Update an existing certificate-key pair by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Certificates** > **Server Certificates**.
2. Select the certificate that you want to update, and click **Update**.


Server Certificates
Install
Update
Delete
Action
Name
Common Name
ns-sftrust-certificate
SFTrust default GMKAEO
storefront.corp.com
storefront.corp.com
WildcardCorpCom
wildcard
ns-server-certificate
Install
Update
Details

3. Select **Update the certificate and key**.


Update Certificate

Certificate-Key Pair Name
storefront.corp.com

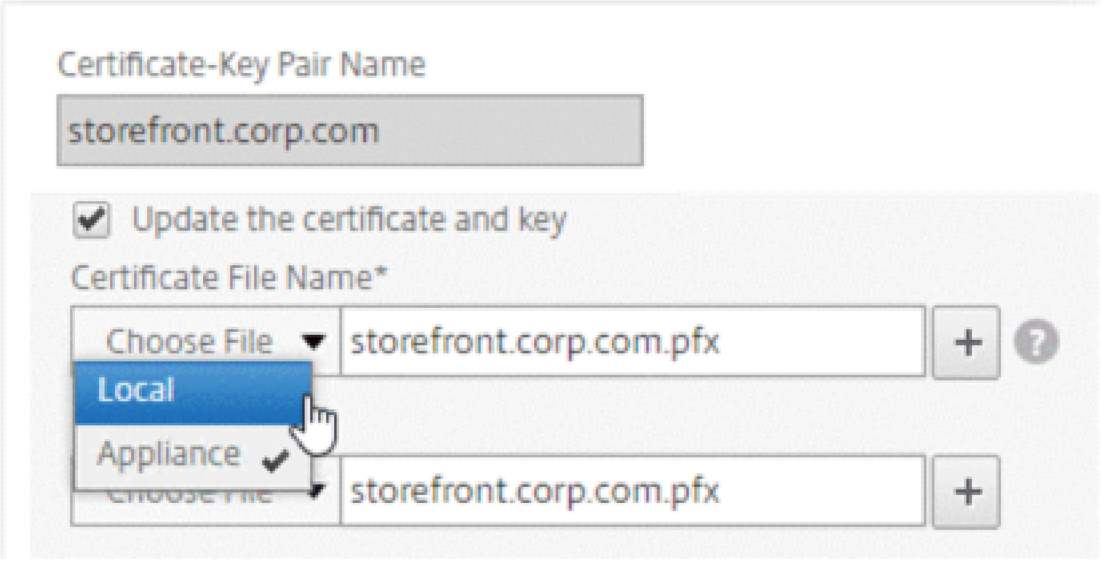
☐ Update the certificate and key

Certificate File Name
storefront.corp.com.pfx

Key Filename
storefront.corp.com.pfx

Certificate Format
PFX

4. In **Certificate File Name**, click **Choose File** > **Local**, and browse to the updated .pfx file or certificate PEM file.


Certificate-Key Pair Name
storefront.corp.com

☑ Update the certificate and key

Certificate File Name*
Choose File: storefront.corp.com.pfx

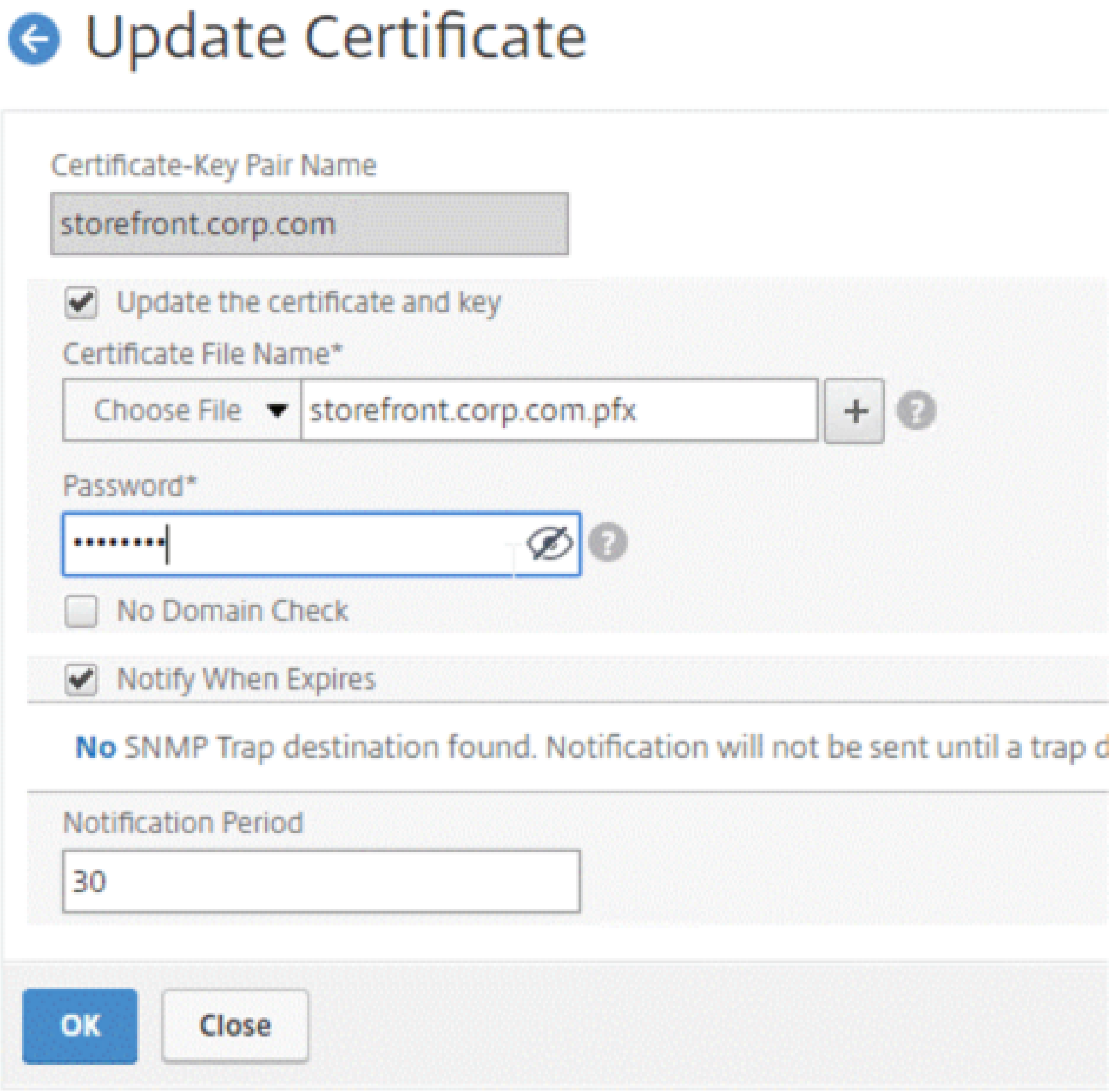
storefront.corp.com.pfx

- If you upload a .pfx file, you are prompted to specify the .pfx file password.
- If you upload a certificate pem file, you must also upload a certificate key file. If the key is encrypted, you must specify the encryption password.

5. If the common name of the new certificate does not match the old certificate, then select **No**

Domain Check.

6. Click **OK**. All the SSL virtual servers to which this certificate is bound are automatically updated.



Certificate-Key Pair Name: storefront.corp.com

☑ Update the certificate and key

Certificate File Name*: Choose File ▾ storefront.corp.com.pfx

Password*: ••••••••

☐ No Domain Check

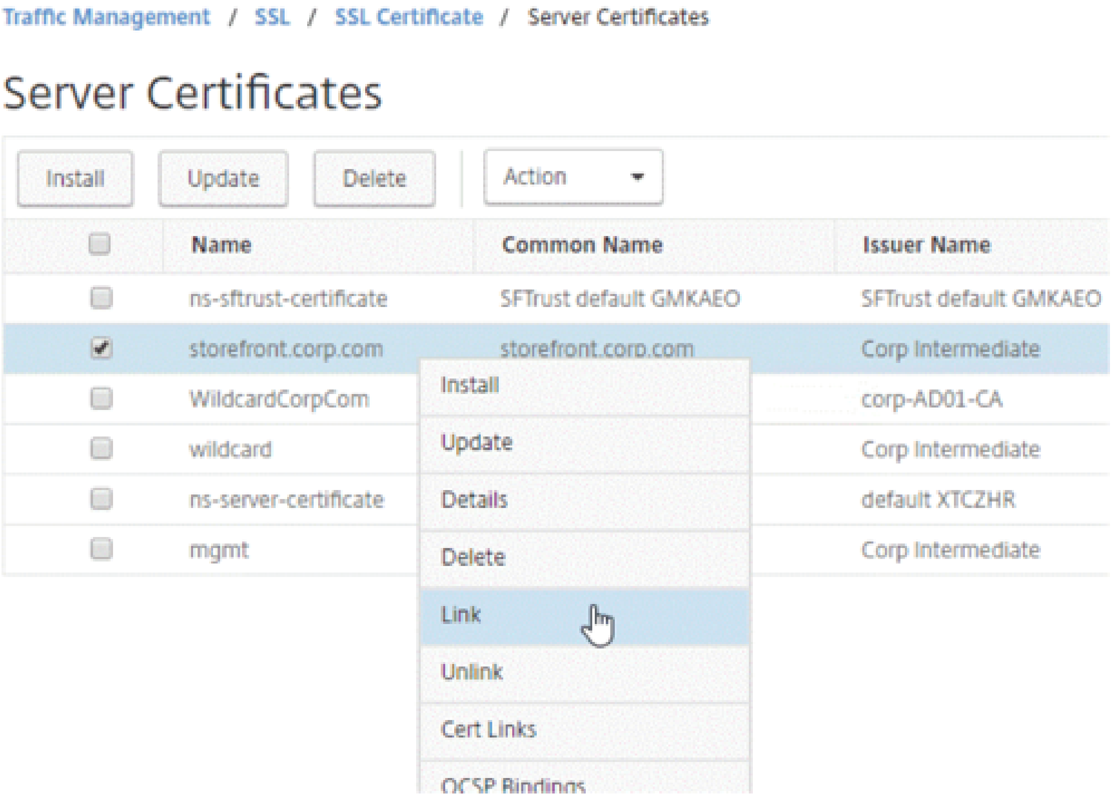
☑ Notify When Expires

No SNMP Trap destination found. Notification will not be sent until a trap d

Notification Period: 30

7. After replacing the certificate, you might have to update the certificate link to a new intermediate certificate. For more information about updating an intermediate certificate without breaking the links, see Update an intermediate certificate without breaking the links.

    - Right-click the updated certificate, and click **Cert Links**, to see if it is linked to an intermediate certificate.
    - If the certificate is not linked, then right-click the updated certificate, and click **Link** to link it to an intermediate certificate. If you don't see an option to link, then you must first have to install a new intermediate certificate on the appliance under the **CA Certificates** node.


Install
Update
Delete
Action
Install
Update
Details
Delete
Link
Unlink
Cert Links
OCSP Bindings

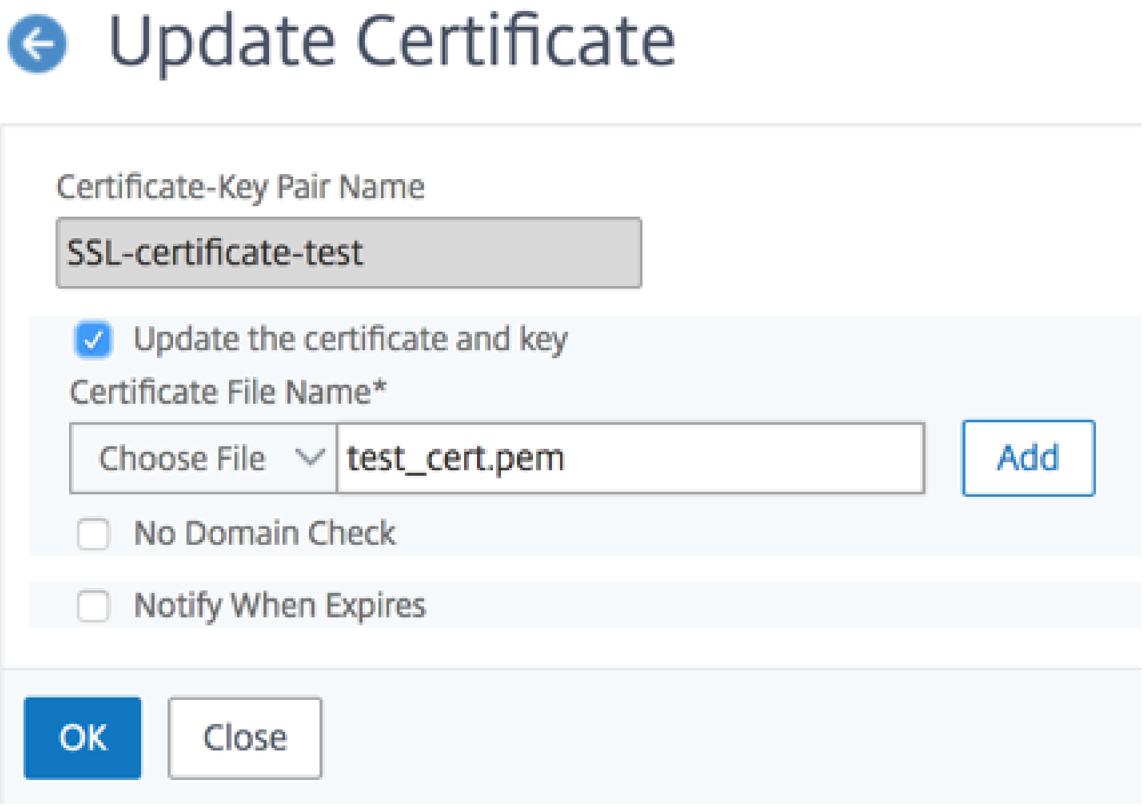
Traffic Management / SSL / SSL Certificate / Server Certificates

Server Certificates

☐	Name	Common Name	Issuer Name
☐	ns-sftrust-certificate	SFTrust default GMKAEO	SFTrust default GMKAEO
☑	storefront.corp.com	storefront.corp.com	Corp Intermediate
☐	WildcardCorpCom		corp-AD01-CA
☐	wildcard		Corp Intermediate
☐	ns-server-certificate		default XTCZHR
☐	mgmt		Corp Intermediate

Update an existing CA certificate

The steps to update an existing CA certificate are the same as updating an existing server certificate. The only difference is that you do not need a key in the case of CA certificates.


Update Certificate
Certificate-Key Pair Name
SSL-certificate-test
☑ Update the certificate and key
Certificate File Name*
Choose File
test_cert.pem
Add
☐ No Domain Check
☐ Notify When Expires
OK
Close

Disable domain checks

When an SSL certificate is replaced on the appliance, the domain name mentioned on the new certificate should match the domain name of the certificate being replaced. For example, if you have a certificate issued to abc.com, and you are updating it with a certificate issued to def.com, the certificate update fails.

However, if you want the server that has been hosting a particular domain to now host a new domain, you can disable the domain check before updating its certificate.

Disable the domain check for a certificate by using the CLI

At the command prompt, type the following commands to disable the domain check and verify the configuration:

```
update ssl certKey <certkeyName> -noDomainCheck

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:

```
update ssl certKey sv -noDomainCheck

Done

show ssl certkey sv

Name: sv

Cert Path: /nsconfig/ssl/complete/server/server_rsa_512.pem

Key Path: /nsconfig/ssl/complete/server/server_rsa_512.ky

Format: PEM

Status: Valid,   Days to expiration:9349

Certificate Expiry Monitor: DISABLED

Done
<!--NeedCopy-->
```

Disable the domain check for a certificate by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates**, select a certificate, and click **Update**.
2. Select **No Domain Check**.

Enable the expiry monitor

An SSL certificate is valid for a specific period. A typical deployment includes multiple virtual servers that process SSL transactions, and the certificates bound to them can expire at different times. An expiry monitor configured on the appliance creates entries in the appliance's syslog and ns audit logs when a certificate configured is due to expire.

If you want to create SNMP alerts for certificate expiration, you must configure them separately.

Enable an expiry monitor for a certificate by using the CLI

At the command prompt, type the following commands to enable an expiry monitor for a certificate and verify the configuration:

```
set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-
    notificationPeriod <positive_integer>]]
```

```

show ssl certKey <certkeyName>
<!--NeedCopy-->
```

Example:

```
set ssl certKey sv -expiryMonitor ENABLED - notificationPeriod 60

Done
<!--NeedCopy-->
```

Enable an expiry monitor for a certificate by using the GUI

1. Navigate to **Traffic Management > SSL > Certificates**, select a certificate, and click **Update**.
2. Select **Notify When Expires**, and optionally specify a notification period.

Update an intermediate certificate without breaking the links

You can now update an intermediate certificate without breaking any existing links. You can do this if the optional 'AuthorityKeyIdentifier' extension, in the linked certificate issued by the certificate to be replaced, does not contain an authority certificate serial number ('authorityCertSerialNumber') field. If the 'AuthorityKeyIdentifier' extension contains a serial number field, then the certificate serial numbers of the old and new certificate must be the same. You can update any number of certificates in the link, one at a time, if the preceding condition is met. Previously, the links broke if an intermediate certificate was updated.

For example, there are four certificates: `CertA`, `CertB`, `CertC`, and `CertD`. `CertA` is the issuer for `CertB`, `CertB` is the issuer for `CertC`, and so on. To replace intermediate certificate `CertB` with `CertB_new`, without breaking the link, the following condition must be met:

The certificate serial number of `CertB` should match the certificate serial number of `CertB_new` if both of the following conditions are met:

- The `AuthorityKeyIdentifier` extension is present in `CertC`.
- This extension contains a serial number field.

If the common name in a certificate changes, while updating the certificate specify `nodomaincheck`.

In the preceding example, to change "www.example.com" in `CertD` to "*.example.com", select the 'No Domain Check" parameter.

Update the certificate by using the CLI

At the command prompt, type:

```
update ssl certKey <certkeyName> -cert <string> [-password] -key <
    string>  [-noDomainCheck]
<!--NeedCopy-->
```

Example:

```
update ssl certkey siteAcertkey -cert /nsconfig/ssl/cert.pem -key /
    nsconfig/ssl/pkey.pem -noDomainCheck
<!--NeedCopy-->
```

Display a certificate chain

A certificate contains the name of the issuing authority and the subject to whom the certificate is issued. To validate a certificate, you must look at the issuer of that certificate and confirm if you trust the issuer. If you do not trust the issuer, you must see who issued the issuer certificate. Go up the chain until you reach the root CA certificate or an issuer that you trust.

As part of the SSL handshake, when a client requests a certificate, the appliance presents a certificate and the chain of issuer certificates that are present on the appliance. An administrator can view the certificate chain for the certificates present on the appliance and install any missing certificates.

View the certificate chain for the certificates present on the appliance by using the CLI

At the command prompt, type:

```
show ssl certchain <cert_name>
<!--NeedCopy-->
```

Examples

There are 3 certificates: c1, c2, and c3. Certificate c1 is signed by c2, c2 is signed by c3, and c3 is the root CA certificate. The following examples illustrate the output of the `show ssl certchain c1` command in different scenarios.

Scenario 1:

Certificate c2 is linked to c1, and c3 is linked to c2.

Certificate c3 is a root CA certificate.

If you run the following command, the certificate links up to the root CA certificate are displayed.

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:

    1) Certificate name: c2               linked; not a root
       certificate

    2) Certificate name: c3               linked; root certificate

Done
<!--NeedCopy-->
```

Scenario 2:

Certificate c2 is linked to c1.

Certificate c2 is not a root CA certificate.

If you run the following command, information that certificate c3 is a root CA certificate but is not linked to c2 is displayed.

```
show ssl certchain c1

Certificate chain  details of certificate name c1 are:

    1) Certificate Name: c2               linked; not a root
       certificate

    2) Certificate Name: c3               not linked; root
       certificate

    Done
<!--NeedCopy-->
```

Scenario 3:

Certificate c1, c2, and c3 are not linked but are present on the appliance.

If you run the following command, information about all the certificates starting with the issuer of certificate c1 is displayed and it is specified that the certificates are not linked.

```
show ssl certchain c1

Certificate chain details of certificate name c1 are:

    1) Certificate Name: c2               not linked; not a root
       certificate

    2) Certificate Name: c3               not linked; root certificate

Done
<!--NeedCopy-->
```

Scenario 4:

Certificate c2 is linked to c1.

Certificate c3 is not present on the appliance.

If you run the following command, information about the certificate linked to c1 is displayed and you are prompted to add a certificate with the subject name specified in c2. In this case, the user is asked to add the root CA certificate c3.

```
show ssl certchain c1

    Certificate chain details of certificate name c1 are:

    1) Certificate Name: c2               linked; not a root certificate

    2) Certificate Name: /C=IN/ST=ka/O=netscaler/CN=test

       Action: Add a certificate with this subject name.

Done
<!--NeedCopy-->
```

Scenario 5:

A certificate is not linked to certificate c1 and the issuer certificate of c1 is not present on the appliance.

If you run the following command, you are prompted to add a certificate with the subject name in certificate c1.

```
sh ssl certchain c1

Certificate chain details of certificate name c1 are:

    1) Certificate Name: /ST=KA/C=IN

       Action: Add a certificate with this subject name.
<!--NeedCopy-->
```

Generate a server test certificate

September 21, 2020

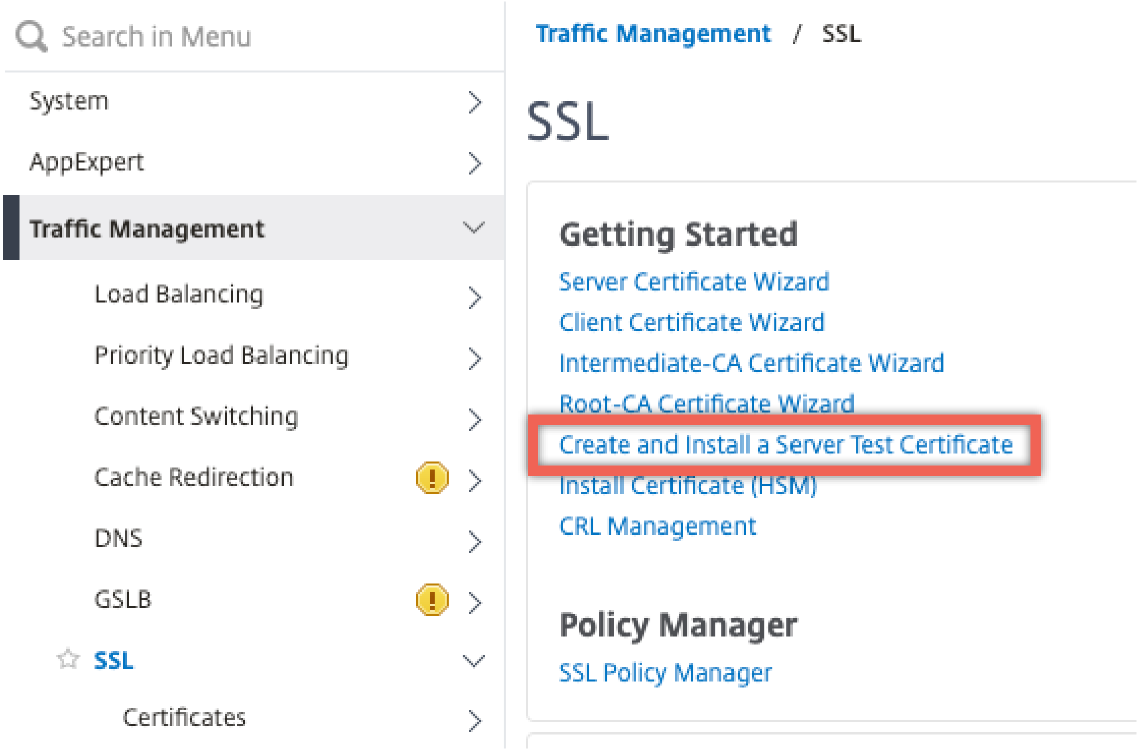
The NetScaler appliance allows you to create a test certificate for server authentication by using a GUI wizard in the configuration utility. A server certificate is used to authenticate and identify a server in an SSL handshake. Typically, a trusted CA issues a server certificate. The server sends the certificate to a client who uses it to authenticate the server.

For issuing a server test certificate, the appliance operates as a certificate authority. This certificate can be bound to an SSL virtual server for authentication in an SSL handshake with a client. This certificate is for testing purposes only. Do not use in a production environment.

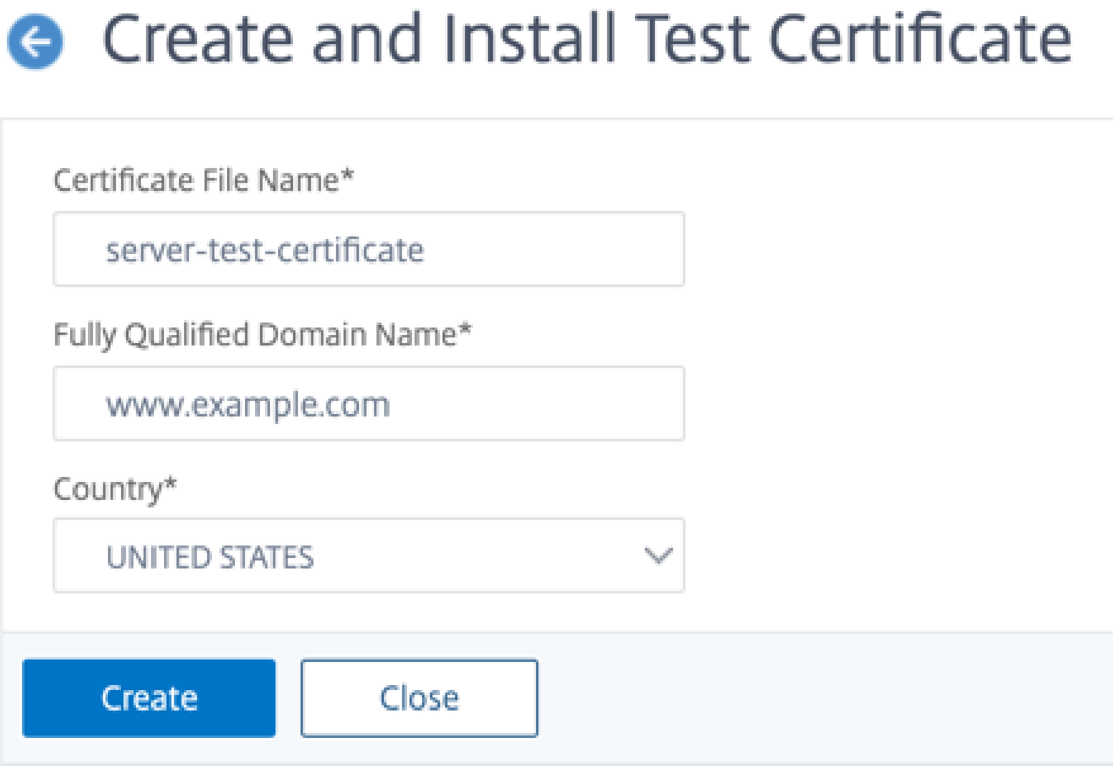
You can install the server test certificate on any virtual server that uses the SSL or the SSL_TCP protocol.

Generate a server test certificate by using the GUI

1. Navigate to **Traffic Management > SSL** and, in the **SSL Certificates group**, select **Create and Install a Server Test Certificate**.


Search in Menu
System
AppExpert
Traffic Management
Load Balancing
Priority Load Balancing
Content Switching
Cache Redirection
DNS
GSLB
SSL
Certificates
Traffic Management / SSL
SSL
Getting Started
Server Certificate Wizard
Client Certificate Wizard
Intermediate-CA Certificate Wizard
Root-CA Certificate Wizard
Create and Install a Server Test Certificate
Install Certificate (HSM)
CRL Management
Policy Manager
SSL Policy Manager

2. Enter details for the parameters and click **Create**.



Create and Install Test Certificate

Certificate File Name*

server-test-certificate

Fully Qualified Domain Name*

www.example.com

Country*

UNITED STATES

Import and convert SSL files

September 21, 2020

You can now import SSL resources, such as certificates, private keys , CRLs, and DH keys, from remote hosts even if FTP access to these hosts is not available. This is especially helpful in environments where shell access to the remote host is restricted. Default folders are created in /nsconfig/ssl as follows:

- For certificate files: /nsconfig/ssl/certfile
- For private keys: the /nsconfig/ssl/keyfile
- For CRLs: /var/netscaler/ssl/crlfile
- For DH keys: /nsconfig/ssl/dhfile

Imports from both HTTP and HTTPS servers are supported. However, the import fails if the file is on an HTTPS server that requires client certificate authentication for access.

> **Note:**
>
> The import command is not stored in the configuration (ns.conf) file, because reimporting the

> file after a restart might cause an error.

Import a certificate file

You can use the CLI and GUI to import a file (resource) from a remote host.

Import a certificate file from a remote host by using the CLI

At the command prompt, type:

```
import ssl certFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl certfile my-certfile http://www.example.com/file_1
<!--NeedCopy-->
```

```
show ssl certfile
     Name : my-certfile
     URL : http://www.example.com/file_1
<!--NeedCopy-->
```

To remove a certificate file, use the rm ssl certFile command, which accepts only the 'name' argument.

Import a key file from a remote host by using the CLI

At the command prompt, type:

```
import ssl keyFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl keyfile my-keyfile http://www.example.com/key_file
<!--NeedCopy-->
```

```
show ssl keyfile
     Name : my-keyfile
     URL : http://www.example.com/key_file
<!--NeedCopy-->
```

To remove a key file, use the rm ssl keyFile command, which accepts only the 'name' argument.

Import a CRL file from a remote host by using the CLI

At the command prompt, type:

```
import ssl crlFile [<name>] [<src>]
<!--NeedCopy-->
```

To remove a CRL file, use the rm ssl crlFile command, which accepts only the <name> argument.

Example:

```
import ssl crlfile my-crlfile http://www.example.com/crl_file

show ssl crlfile

    Name : my-crlfile

    URL : http://www.example.com/crl_file
<!--NeedCopy-->
```

Import a DH file from a remote host by using the CLI

At the command prompt, type:

```
import ssl dhFile [<name>] [<src>]
<!--NeedCopy-->
```

Example:

```
import ssl dhfile my-dhfile http://www.example.com/dh_file
show ssl dhfile
     Name : my-dhfile
     URL : http://www.example.com/dh_file
<!--NeedCopy-->
```

To remove a DH file, use the rm ssl dhFile command, which accepts only the <name> argument.

Import an SSL resource by using the GUI

Navigate to Traffic Management > SSL > Imports, and then select the appropriate tab.

Import PKCS#8 and PKCS#12 certificates

If you want to use certificates and keys that you already have on other secure servers or applications in your network, you can export them, and then import them to the NetScaler appliance. You might have to convert exported certificates and keys before you can import them to the NetScaler appliance.

For the details of how to export certificates from secure servers or applications in your network, see the documentation of the server or application from which you want to export.

> **Note:**
>
> For installation on the NetScaler appliance, key and certificate names cannot contain spaces or special characters other than those supported by the UNIX file system. Follow the appropriate naming convention when you save the exported key and certificate.

A certificate and private key pair is commonly sent in the PKCS#12 format. The appliance supports PEM and DER formats for certificates and keys. To convert PKCS#12 to PEM or DER, or PEM or DER to PKCS#12, see the "Convert SSL certificates for import or export" section later in this page.

The NetScaler appliance does not support PEM keys in PKCS#8 format. However, you can convert these keys to a supported format by using the OpenSSL interface, which you can access from the CLI or the configuration utility. Before you convert the key, you need to verify that the private key is in PKCS#8 format. Keys in PKCS#8 format typically start with the following text:

```
-----BEGIN ENCRYPTED PRIVATE KEY-----



leuSSZQZKgrgUQ==



-----END ENCRYPTED PRIVATE KEY-----
<!--NeedCopy-->
```

Open the OpenSSL interface from the CLI

1. Open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance by using the administrator credentials.
3. At the command prompt, type shell.
4. At the shell prompt type openssl.

Open the OpenSSL interface from the GUI

Navigate to Traffic Management > SSL and, in the Tools group, select OpenSSL interface.

Convert a non-supported PKCS#8 key format to an encrypted supported key format by using the OpenSSL interface

At the OpenSSL prompt, type one of the following commands, depending on whether the non-supported key format is of type rsa or dsa:

```
OpenSSL>rsa- in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>

OpenSSL>dsa -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>
<!--NeedCopy-->
```

Parameters for converting an unsupported key format to a supported key format

- **PKCS#8 Key Filename:** The input file name of the incompatible PKCS#8 private key.
- **encrypted Key Filename:** The output file name of the compatible encrypted private key in PEM format.
- **unencrypted Key Filename:** The output file name of the compatible unencrypted private key in PEM format.

Convert SSL certificates for import or export

A NetScaler appliance supports the PEM and DER formats for SSL certificates. Other applications, such as client browsers and some external secure servers, require various public key cryptography standard (PKCS) formats. The NetScaler appliance can convert the PKCS#12 format (the personal information exchange syntax standard) to PEM or DER format for importing a certificate to the appliance, and can convert PEM or DER to PKCS#12 for exporting a certificate. For additional security, conversion of a file for import can include encryption of the private key with the DES or DES3 algorithm.

> **Note:**
>
> If you use the GUI to import a PKCS#12 certificate, and the password contains a dollar sign ($), backquote (‘), or escape () character, the import may fail. If it does, the ERROR: Invalid password message appears. If you must use a special character in the password, be sure to prefix it with an escape character () unless all imports are performed by using the CLI.

Convert the format of a certificate by using the CLI

At the command prompt, type the following command:

```
convert ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-
    des | -des3] [-export [-certFile <inputFilename>] [-keyFile <
    inputFilename>]]
<!--NeedCopy-->
```

During the operation, you are prompted to enter an import password or an export password. For an encrypted file, you are also prompted to enter a passphrase.

Example:

```
convert ssl pkcs12 Cert-Import-1.pem -import -pkcs12File Cert-Import-1.
    pfx -des

convert ssl pkcs12 Cert-Client-1.pfx -export -certFile Cert-Client-1 -
    keyFile Key-Client-1
<!--NeedCopy-->
```

Convert the format of a certificate by using the GUI

Navigate to Traffic Management > SSL and, in the Tools group, select Import PKCS#12 or Export PKCS#12.

SSL profiles

September 21, 2020

An SSL profile is a collection of settings for SSL entities. It offers ease of configuration and flexibility. Instead of configuring the settings on each entity, you can configure them in a profile and bind the profile to all the entities that the settings apply to.

The SSL profile infrastructure has been enhanced to use the latest ciphers and protocols. Differences between the legacy profile (old profile) and the enhanced SSL profile (new profile) are highlighted.

Differences between the old and the new SSL profile infrastructure

Differences	Old Profile	New Profile
Ciphers and ECC Curves included in the profile	No	Yes

Differences	Old Profile	New Profile
Inserting a cipher or cipher group in the middle of an existing list	Unbind all the ciphers and bind again in the order of the required priority.	Add a cipher and assign it a priority. If a priority is not specified, the cipher is assigned the lowest priority in the list.
Unbinding all the ciphers	> unbind ssl vserver <name> ciphername –ALL	unbind ssl profile –cipherName FlushAllCiphers (Release 11.0 build 64.x or later includes the FlushAllCiphers parameter for unbinding all the ciphers or cipher groups from a profile, because ALL is treated like a cipher group.)
State of SSLv3	n/a	Disabled on the default front-end profile (ns_default_ssl_profile_frontend). Note: Before you enable this profile, SSLv3 is enabled globally. After enabling the profile, SSLv3 is disabled on the front-end default profile.

SSL profile infrastructure

September 21, 2020

Vulnerabilities in SSLv3 and RC4 implementation have emphasized the need to use the latest ciphers and protocols to negotiate the security settings for a network connection. Implementing any changes to the configuration, such as disabling SSLv3 across thousands of SSL end points, is a cumbersome process. Therefore, settings that were part of the SSL end points configuration have been moved to the SSL profiles, along with the default ciphers. To implement changes in the configuration, including cipher support, you need only modify the profile that is bound to the entities.

The default SSL profiles (default front-end and default back-end) contain all the default ciphers and ECC curves, in addition to the settings that were part of the old profiles. Sample outputs for the default profiles are provided in the appendix. The Enable Default Profile operation automatically binds the default front-end profile to all front-end entities, and the default back-end profile to all back-end entities. You can modify a default profile to suit your deployment. You can also create custom profiles and bind them to SSL entities.

> **Important:**
>
> After the upgrade, if you enable the default profiles, you cannot undo the changes. That is, the profiles cannot be disabled. Save the configuration and create a copy of the configuration file (ns.conf) before enabling the profiles. However, if you do not want to use the features in the default profile, you can continue to use the old SSL profiles. For more information about these profiles, see Legacy SSL profile.
>
> From release 11.1 51.x, in the GUI and CLI, a confirmation prompt is added when you enable the default profile to avoid enabling it by mistake.

Command:

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
       cannot undo the changes. Are you sure you want to enable the
       Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

By default, some SSL parameters, called *global parameters*, apply to all the SSL end points. However, if a profile is bound to an SSL end point, the global parameters do not apply. The settings specified in the profile apply instead.

Points to note

1. A profile can be bound to multiple virtual servers, but a virtual server can have only one profile bound to it.
2. To delete a profile that is bound to a virtual server, first unbind the profile.
3. A cipher or cipher group can be bound to multiple profiles at different priorities.
4. A profile can have multiple ciphers and cipher groups bound at different priorities.
5. Changes to a cipher group are immediately reflected in all the profiles and in all the virtual servers that one of the profiles is bound to.
6. If a cipher suite is part of a cipher group, to remove the cipher suite from the profile first edit the cipher group to remove the specific cipher suite.
7. If you do not assign a priority to a cipher suite or cipher group that you attach to a profile, it is assigned the lowest priority within the profile.
8. You can create a custom cipher group (also called a user-defined cipher group) from existing cipher groups and cipher suites. If you create cipher group A and add existing cipher groups X

and Y to it, in that order, cipher group Y is assigned at a lower priority than cipher group X. That is, the group that is added first has a higher priority.

9. If a cipher suite is already part of a cipher group that is attached to a profile, and the same cipher suite is part of another cipher group that is also attached to the same profile, the cipher suite is not added again as part of the second cipher group. The cipher suite at the higher priority is in effect when traffic is processed.
10. Cipher groups are not expanded in the profile. As a result, the number of lines in the configuration file (ns.conf) is greatly reduced. For example, if there are a thousand SSL virtual servers to which two cipher groups are bound, and each cipher group contains 15 ciphers, expansion would result in 30*1000 entries related to ciphers in the configuration file. With the new profile, it would have only two entries: one for each cipher group that is bound to a profile.
11. Creating a user defined cipher group from existing ciphers and cipher groups is a copy-paste operation. Any changes in the original group are not reflected in the new group.
12. A user-defined cipher group lists all the profiles that it is a part of.
13. A profile lists all the SSL virtual server, services, and service groups that it is bound to.
14. If the default SSL profile feature is enabled, you must use the profile to set or change any of the attributes of a virtual server, service, service group, or an internal service.

Save the configuration by using the CLI

At the command prompt, type:

```
save config

shell

root@ns# cd /nsconfig

root@ns# cp ns.conf ns.conf.NS<currentreleasenumber><currentbuildnumber>
<!--NeedCopy-->
```

Example:

```
save config
shell
root@ns# cd /nsconfig
root@ns# cp ns.conf ns.conf.NS.11.0.jun.16
<!--NeedCopy-->
```

Enable the default profile

> **Important:**
>
> Save your configuration before you upgrade the software and enable the default profiles.
>
> From release 11.1 build 51.x, in the GUI and CLI, a confirmation prompt appears when you enable the default profile to avoid enabling it by mistake.

Command: The following command enables the default profile and binds this profile to the SSL entities to which a profile is already bound. That is, if a profile (for example P1) is already bound to an SSL entity, P1 is replaced by the default front-end profile or the default back-end profile. The older profile (P1) is not deleted. It is now an enhanced SSL profile and contains the earlier settings, and the ciphers and ECC curves. If you do not want the default profile, you can explicitly bind P1 to the SSL entity.

```
set ssl parameter -defaultProfile ENABLED
    Save your configuration before enabling the Default profile. You
        cannot undo the changes. Are you sure you want to enable the
        Default profile? [Y/N]Y
Done
<!--NeedCopy-->
```

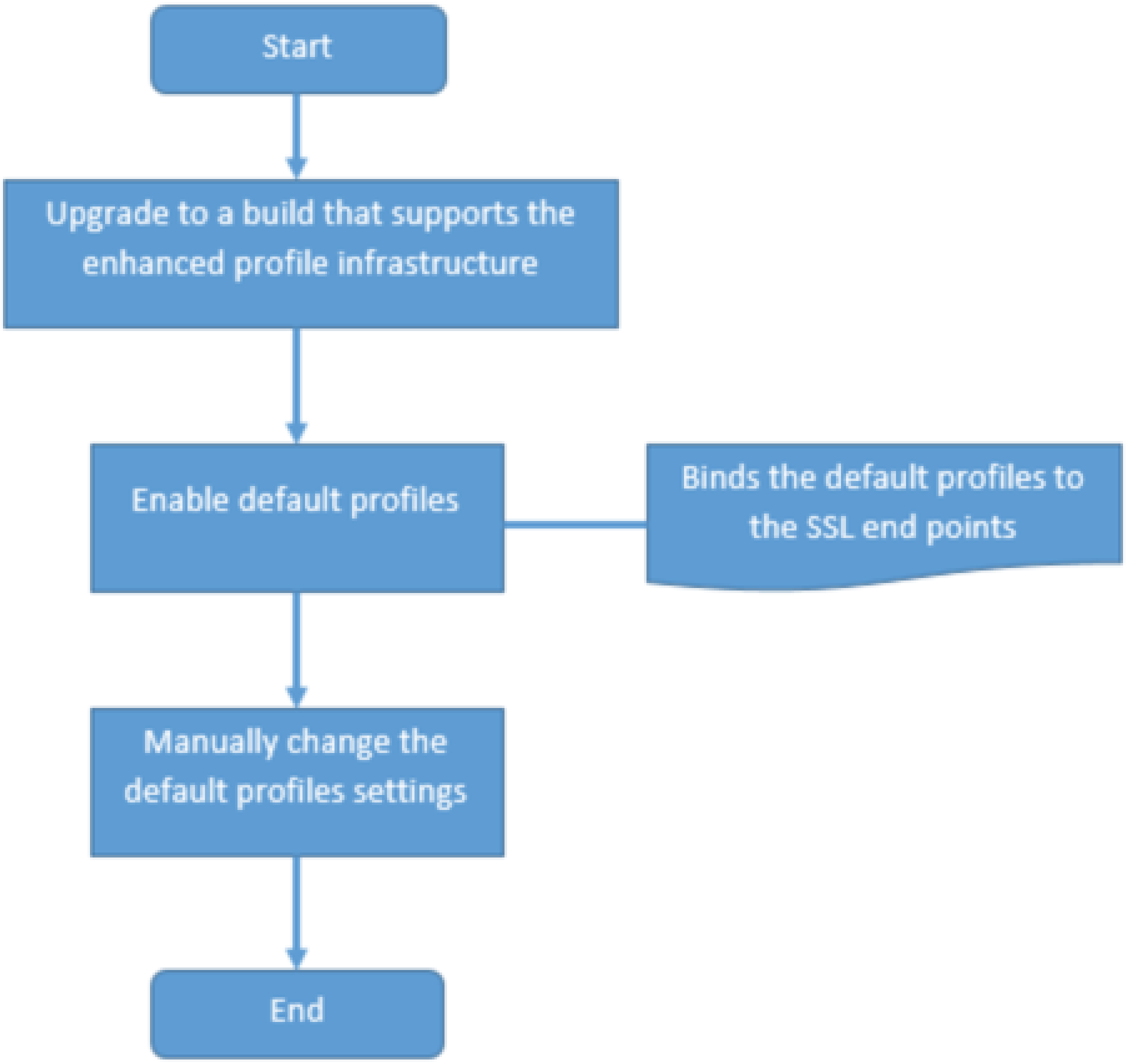
Upgrade the software to a build that supports the enhanced profile infrastructure, and then enable the default profiles.

> **Notes:**
>
> - If a legacy profile (P1) is already bound to an SSL entity, and you enable the default profile, the default profile overrides the earlier binding. That is, the default profile is bound to the SSL entities. If you do not want the default profile to be bound, you must bind P1 to the SSL entity again.
> - A single operation (Enable Default Profile or `set ssl parameter -defaultProfile ENABLED`) enables (binds) both the default front-end profile and the default back-end profile.

Use case

After you enable the default profiles, they are bound to all the SSL end points. The default profiles are editable. If your deployment uses most of the default settings and changes only a few parameters, you can edit the default profiles. The changes are immediately reflected across all the end points. You can also create custom SSL profiles with some custom and some default parameters and bind it to the SSL entities.

The following flowchart explains the steps that you must perform:


Start
Upgrade to a build that supports the enhanced profile infrastructure
Enable default profiles
Binds the default profiles to the SSL end points
Manually change the default profiles settings
End

1. For information about upgrading the software, see Upgrading the System Software.
2. Enable the default profiles by using the CLI or GUI.

- At the command line, type: `set ssl parameter -defaultProfile ENABLED`
- If you prefer to use the GUI, navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**, scroll down, and select **Enable Default Profile**.

If a profile was not bound to an end point before the upgrade, a default profile is bound to the SSL end point. If a profile was bound to an end point before the upgrade, the same profile is bound after the upgrade, and default ciphers are added to the profile.

1. (Optional) Manually change any settings in the default profile.

- At the command line, type: `set ssl profile <name>` followed by the parameters to modify.

- If you prefer to use the GUI, navigate to **System** > **Profiles**. In **SSL Profiles**, select a profile and click **Edit**.

SSL profile parameters

You can set the following SSL parameters in an SSL profile. You can set some of these parameters in an SSL virtual server. For more information about SSL virtual server parameters, see SSL virtual server parameters.

Set SSL profile parameters by using the CLI

At the command prompt, type:

```
set ssl profile <name> [-ssllogProfile <string>] [-dh ( ENABLED |

      DISABLED ) -dhFile <string>] [-dhCount <positive_integer>]

      [-dhKeyExpSizeLimit ( ENABLED | DISABLED )] [-eRSA ( ENABLED | DISABLED

      ) [-eRSACount <positive_integer>]] [-sessReuse ( ENABLED | DISABLED )

      [-sessTimeout <positive_integer>]] [-cipherRedirect ( ENABLED |

      DISABLED ) [-cipherURL <URL>]] [-clientAuth ( ENABLED | DISABLED )

      [-clientCert ( Mandatory | Optional )]] [-sslRedirect ( ENABLED |

      DISABLED )] [-redirectPortRewrite ( ENABLED | DISABLED )] [-ssl3 (

      ENABLED | DISABLED )] [-tls1 ( ENABLED | DISABLED )] [-tls11 ( ENABLED

      | DISABLED )] [-tls12 ( ENABLED | DISABLED )] [-tls13 ( ENABLED |

      DISABLED )] [-SNIEnable ( ENABLED | DISABLED )] [-ocspStapling (

      ENABLED | DISABLED )] [-serverAuth ( ENABLED | DISABLED )] [-commonName

      <string>] [-pushEncTrigger <pushEncTrigger>] [-sendCloseNotify ( YES |

      NO )] [-clearTextPort <port|*>] [-insertionEncoding ( Unicode | UTF-8

      )] [-denySSLReneg <denySSLReneg>] [-quantumSize <quantumSize>]

      [-strictCAChecks ( YES | NO )] [-encryptTriggerPktCount

      <positive_integer>] [-pushFlag <positive_integer>]

      [-dropReqWithNoHostHeader ( YES | NO )] [-pushEncTriggerTimeout

      <positive_integer>] [-sslTriggerTimeout <positive_integer>]

      [-clientAuthUseBoundCAChain ( ENABLED | DISABLED )] [-sslInterception (

      ENABLED | DISABLED )] [-ssliReneg ( ENABLED | DISABLED )]

      [-ssliOCSPCheck ( ENABLED | DISABLED )] [-ssliMaxSessPerServer

      <positive_integer>] [-HSTS ( ENABLED | DISABLED )] [-maxage

      <positive_integer>] [-IncludeSubdomains ( YES | NO )] [-sessionTicket (

      ENABLED | DISABLED )] [-sessionTicketLifeTime <positive_integer>]

      [-sessionTicketKeyRefresh ( ENABLED | DISABLED )]

      {
-sessionTicketKeyData  }
 [-sessionKeyLifeTime <positive_integer>]

      [-prevSessionKeyLifeTime <positive_integer>] [-cipherName <string>

      -cipherPriority <positive_integer>] [-strictSigDigestCheck ( ENABLED |

      DISABLED )] [-skipClientCertPolicyCheck ( ENABLED | DISABLED )]
```

```

        [-zeroRttEarlyData ( ENABLED | DISABLED )]

        [-tls13SessionTicketsPerAuthContext <positive_integer>]

        [-dheKeyExchangeWithPsk ( YES | NO )]
<!--NeedCopy-->
```

TLS session ticket extension

An SSL handshake is a CPU-intensive operation. If session reuse is enabled, the server/client key exchange operation is skipped for existing clients. They are allowed to resume their sessions. This action improves the response time and increases the number of SSL transactions per second that a server can support. However, the server must store details of each session state, which consumes memory and is difficult to share among multiple servers if requests are load balanced across servers.

NetScaler appliances support the SessionTicket TLS extension. Use of this extension indicates that the session details are stored on the client instead of on the server. The client must indicate that it supports this mechanism by including the session ticket TLS extension in the client Hello message. For new clients, this extension is empty. The server sends a new session ticket in the NewSessionTicket handshake message. The session ticket is encrypted by using a key-pair known only to the server. If a server cannot issue a new ticket now, it completes a regular handshake.

This feature is available only in front-end SSL profiles, and only at the front end of communication in which the NetScaler appliance acts as a server and generates session tickets.

Limitations

- This feature is not supported on a FIPS platform.
- This feature is supported only with TLS versions 1.1 and 1.2.
- SSL session ID persistency is not supported with session tickets.

Enable TLS session ticket extension by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket (ENABLED | DISABLED ) [-
    sessionTicketLifeTime <positive_integer>
<!--NeedCopy-->
```

Arguments:

sessionTicket: State of TLS session ticket extension. Use of this extension indicates that the session details are stored on the client instead of on the server, as defined in RFC 5077.

Possible values: ENABLED, DISABLED

Default value: DISABLED

sessionTicketLifeTime: Specify a time, in seconds, after which the session ticket expires and a new SSL handshake must be initiated.

Default value: 300

Minimum value: 0

Maximum value: 172800

Example:

```
add ssl profile profile1 -sessionTicket ENABLED -sessionTicketlifeTime 300
Done
<!--NeedCopy-->
```

Enable TLS session ticket extension by using the GUI

1. Navigate to **System** > **Profiles**. Select **SSL Profiles**.
2. Click **Add** and specify a name for the profile.
3. Select **Session ticket**.
4. Optionally, specify **Session Ticket Lifetime (secs)**.

Secure implementation of session tickets

By using TLS session tickets, clients can use abbreviated handshakes for faster reconnection to servers. However, if session tickets aren't encrypted or changed for long periods of time, they can pose a security risk. You can secure session tickets by encrypting them with a symmetric key. To achieve forward secrecy, you can specify a time interval at which the session-ticket key is refreshed.

Session ticket keys are generated by the appliance by default. However, if multiple appliances in a deployment need to decrypt each other's session tickets, they must all use the same session-ticket key. Therefore, you must set (add or load) the same session-ticket key data manually on all the appliances. Session-ticket key data includes the following information:

- Session ticket name.
- Session AES key used to encrypt or decrypt the ticket.
- Session HMAC key used to compute the digest of the ticket.

You can now configure session ticket key data of length 64 bytes to support 256-bit HMAC keys as recommended in RFC 5077. Key lengths of 48 bytes are also supported for backward compatibility.

> **Note:**
>
> If you are typing the session-ticket key data manually, ensure that the configuration across all the NetScaler appliances in an HA setup or in a cluster setup is the same.

The sessionTicketKeyLifeTime parameter specifies how often a session-ticket key is refreshed. In addition, you can set the prevSessionTicketKeyLifeTime parameter to specify how long the previous session-ticket key will be maintained for decrypting tickets using that key, after a new key is generated. The prevSessionTicketKeyLifeTime setting extends the time for which a client can use an abbreviated handshake to reconnect. For example, if sessionTicketKeyLifeTime is set to 10 minutes and prevSessionTicketKeyLifeTime to 5 minutes, a new key is generated after 10 minutes and used for all new sessions. However, previously connected clients have another 5 minutes for which previously issued tickets will be honored for an abbreviated handshake.

Configure SSL session-ticket data by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket ENABLED -sessionTicketLifeTime <positive_integer> -sessionTicketKeyRefresh ( ENABLED | DISABLED )] -sessionTicketKeyLifeTime <positive_integer> [-prevSessionTicketKeyLifeTime <positive_integer>]
<!--NeedCopy-->
```

Arguments:

sessionTicket: Use session tickets as described by RFC 5077. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the **ENABLED** setting, a server issues a session ticket to a client, which the client can use to perform an abbreviated handshake.

Possible values: ENABLED, DISABLED. Default: DISABLED

sessionTicketLifeTime: Lifetime, in seconds, of the session ticket. After this time expires, clients cannot use this ticket to resume their sessions.

Maximum value: 172800. Minimum value: 0. Default: 300.

sessionTicketKeyRefresh: When the time specified by the session-ticket key lifetime parameter expires, regenerate the session-ticket key used to encrypt or decrypt the session tickets. Automatically enabled if sessionTicket is enabled. Disabled if the session-ticket data is entered by an administrator.

Possible values: ENABLED, DISABLED. Default: ENABLED

sessionKeyLifeTime: Lifetime, in seconds, of a symmetric key used to encrypt the session tickets issued by a NetScaler appliance.

Maximum value: 86400. Minimum value: 600. Default: 3000

prevSessionKeyLifeTime: Time, in seconds, for which the previous symmetric key used to encrypt session tickets remains valid for existing clients after the session-ticket key lifetime expires. Within this time, existing clients can resume their sessions by using the previous session ticket key. Session tickets for new clients are encrypted by using the new key.

Maximum value: 172800. Minimum value: 0. Default: 0

Example:

```
set ssl profile ns_default_ssl_profile_frontend -sessionTicket ENABLED
    -sessionTicketlifeTime 120 -sessionTicketKeyRefresh ENABLED -
    sessionTicketKeyLifeTime 100 -prevSessionTicketKeyLifeTime 60

Done

show ssl profile ns_default_ssl_profile_frontend

    Session Ticket: ENABLED

    Session Ticket Lifetime: 120 (secs)

    Session Key Auto Refresh: ENABLED

    Session Key Lifetime: 100 (secs)

    Previous Session Key Lifetime: 60 (secs)
<!--NeedCopy-->
```

Configure SSL session-ticket data by using the GUI

1. Navigate to **System** > **Profiles**, and select **SSL Profile**.
2. Select **ns_default_ssl_profile_frontend** and click **Edit**.
3. In the **Basic Settings** section, click the pencil icon and set the following parameters:
    - Session Ticket
    - Session Ticket Lifetime (secs)
    - Session Ticket Key Auto Refresh
    - Session Ticket Key Lifetime (secs)
    - Previous Session Ticket Key Lifetime (secs)
4. Click **OK**.

Type SSL session ticket data manually by using the CLI

At the command prompt, type:

```
set ssl profile <name> -sessionTicket ENABLED

set ssl profile <name> -sessionTicketKeyData

show ssl profile ns_default_ssl_profile_frontend
<!--NeedCopy-->
```

Arguments:

sessionTicket: Use of session tickets as described by RFC 5077. Establishing the initial handshake requires CPU-intensive public key encryption operations. With the **ENABLED** setting, a server issues a session ticket to a client, which the client can use to perform an abbreviated handshake.

Possible values: ENABLED, DISABLED. Default: DISABLED

sessionTicketKeyData: Contains the session ticket name (0–15 bytes), the session AES key used to encrypt or decrypt the session ticket (16–31 bytes), and the session HMAC key used to compute the digest of the ticket (32–63 bytes). Externally generated by an administrator and added to a NetScaler appliance.

Maximum length: 64 bytes

Example:

```
set ssl profile ns_default_ssl_profile_frontend -sessionticket ENABLED

Done

set ssl profile ns_default_ssl_profile_frontend -sessionticketkey 111111111111111111111111111111111111111111111111

Done

show ssl profile ns_default_ssl_profile_frontend

    1) Name: ns_default_ssl_profile_frontend (Front-End)

    SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED

    Client Auth: DISABLED

    Use only bound CA certificates: DISABLED

    Strict CA checks: NO

    Session Reuse: ENABLED Timeout: 120 seconds

    DH: DISABLED

    DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED
        Refresh Count: 0

    Deny SSL Renegotiation ALL

    Non FIPS Ciphers: DISABLED

    Cipher Redirect: DISABLED

    SSL Redirect: DISABLED

    Send Close-Notify: YES

    Push Encryption Trigger: Always

    PUSH encryption trigger timeout: 1 ms

    SNI: DISABLED

    OCSP Stapling: DISABLED

    Strict Host Header check for SNI enabled SSL sessions: NO

    Push flag: 0x0 (Auto)

    SSL quantum size: 8 kB

    Encryption trigger timeout 100 mS

    Encryption trigger packet count: 45

    Subject/Issuer Name Insertion Format: Unicode

    Session Ticket: ENABLED

    Session Ticket Lifetime: 300 (secs)

    Session Key Auto Refresh: DISABLED

    Session Key Lifetime: 3000 (secs)

    Previous Session Key Lifetime: 0 (secs)

    Session Key Data: 84
        dad1afc6d56b0deeb0a7fd7f299a207e8d8c15cdd087a5684a11a329fd732e87a0535d9088


    47
        e8c181ba266f5c8838ae472cb3ab9255b683bf922fad32cee816c329989ef7cdeb278e93ac


    ECC Curve: P_256, P_384, P_224, P_521

    1) Cipher Name: DEFAULT Priority :4

    Description: Predefined Cipher Alias

    1) Internal Service Name (Front-End): nsrnatsip-127.0.0.1-5061

    2) Internal Service Name (Front-End): nskrpcs-127.0.0.1-3009

    3) Internal Service Name (Front-End): nshttps-::1l-443

    4) Internal Service Name (Front-End): nsrpcs-::1l-3008

    5) Internal Service Name (Front-End): nshttps-127.0.0.1-443

    6) Internal Service Name (Front-End): nsrpcs-127.0.0.1-3008

    7) Vserver Name: v1

Done
<!--NeedCopy-->
```

Type SSL session ticket data manually by using the GUI

1. Navigate to **System** > **Profiles**, and select **SSL Profile**.
2. Select **ns_default_ssl_profile_frontend** and click **Edit**.
3. In the **Basic Settings** section, click the pencil icon and set the following parameters:
    - Session Ticket
    - Session Ticket Key Data

    - Confirm Session Ticket Key Data

4. Click **OK**.

Load an old configuration

Enabling the default profiles is not reversible. However, if you decide that your deployment does not require the default profiles, you can load an older configuration that you saved before you enabled the default profiles. The changes are effective after you restart the appliance.

Load an old configuration by using the CLI

At the command prompt, type:

```
shell

root@ns# clear config

root@ns# cd /nsconfig

root@ns# cp ns.conf.NS.11.0.jun.16 ns.conf

root@ns# reboot
<!--NeedCopy-->
```

Appendix A: Sample migration of the SSL configuration after upgrade

September 21, 2020

Note: This content has been removed because the SSL migration script for the new default profile is no longer supported.

Appendix B: Default front-end and back-end SSL profile settings

September 21, 2020

A default front-end profile has the following settings:

```
sh ssl profile ns_default_ssl_profile_frontend

```

```
1)Name: ns_default_ssl_profile_frontend

     Configuration for Front-End SSL profile

     DH: DISABLED

     Ephemeral RSA: ENABLED          Refresh Count: 0

     Session Reuse: ENABLED          Timeout: 120 seconds

     Non FIPS Ciphers: DISABLED

     Cipher Redirect: ENABLED   Redirect URL: http://10.102.28.212/
   redirect.html

     Client Auth: DISABLED

     SSL Redirect: DISABLED

     SNI: DISABLED

     SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED  TLSv1.2:
   ENABLED

     Push Encryption Trigger: Always

     PUSH encryption trigger timeout:     1 ms

     Send Close-Notify: YES

     Push flag: 0x0 (Auto)

     Deny SSL Renegotiation          NO

     SSL quantum size:          8 kB

     Strict CA checks:          NO

     Encryption trigger timeout 100 mS

     Encryption trigger packet count:     45

     Use only bound CA certificates: DISABLED

     Subject/Issuer Name Insertion Format: Unicode

     Strict Host Header check for SNI enabled SSL sessions:          NO


     ECC Curve: P_256, P_384, P_521


1)   Cipher Name: AES     Priority :2

     Description: Predefined Cipher Alias


1)   Vserver Name: v1

2)   Vserver Name: nshttps-::1l-443

3)   Vserver Name: nsrpcs-::1l-3008

4)   Vserver Name: nskrpcs-127.0.0.1-3009

5)   Vserver Name: nshttps-127.0.0.1-443

6)   Vserver Name: nsrpcs-127.0.0.1-3008

Done
<!--NeedCopy-->
```

A default back-end profile has the following settings:

```
sh ssl profile ns_default_ssl_profile_backend

1)Name: ns_default_ssl_profile_backend

     Configuration for Back-End SSL profile

     Session Reuse: ENABLED          Timeout: 300 seconds

     Non FIPS Ciphers: DISABLED

     Server Auth: DISABLED

     SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED  TLSv1.2: DISABLED

     Push Encryption Trigger: Always

     PUSH encryption trigger timeout:     1 ms

     Send Close-Notify: YES

     Push flag: 0x0 (Auto)

     Deny SSL Renegotiation          ALL

     SSL quantum size:          8 kB

     Strict CA checks:          NO

     Encryption trigger timeout 100 mS

     Encryption trigger packet count:     45

     Use only bound CA certificates: DISABLED


     ECC Curve: P_256, P_224, P_521


1)   Cipher Name: AES     Priority :1

     Description: Predefined Cipher Alias


2)   Cipher Name: RC4     Priority :2

     Description: Predefined Cipher Alias


1)   Service Name: s2

2)   Service Name: s1

Done
<!--NeedCopy-->
```

Legacy SSL profile

September 21, 2020

> **Note:**
>
> Citrix recommends using the enhanced profiles instead of legacy profiles. For information about the enhanced profile infrastructure, see SSL profile infrastructure.

Important:

> You must bind an SSL profile to an SSL virtual server. Do not bind a DTLS profile to an SSL virtual server. For information about DTLS profiles, see DTLS Profiles.

You can use an SSL profile to specify how a NetScaler processes SSL traffic. The profile is a collection of SSL parameter settings for SSL entities, such as virtual servers, services, and service groups, and offers ease of configuration and flexibility. You are not limited to configuring only one set of global parameters. You can create multiple sets (profiles) of global parameters and assign different sets to different SSL entities. SSL profiles are classified into two categories:

- Front end profiles, containing parameters applicable to the front-end entity. That is, they apply to the entity that receives requests from a client.
- Backend profiles, containing parameters applicable to the back-end entity. That is, they apply to the entity that sends client requests to a server.

Unlike a TCP or HTTP profile, an SSL profile is optional. Therefore, there is no default SSL profile. The same profile can be reused across multiples entities. If an entity does not have a profile attached, the values set at the global level apply. For dynamically learned services, current global values apply.

The following table lists the parameters that are part of each profile.

Front end profile	Backend profile
cipherRedirect, cipherURL	denySSLReneg
clearTextPort*	encryptTriggerPktCount
clientAuth, clientCert	nonFipsCiphers
denySSLReneg	pushEncTrigger
dh, dhFile, dhCount	pushEncTriggerTimeout
dropReqWithNoHostHeader	pushFlag
encryptTriggerPktCount	quantumSize
eRSA, eRSACount	serverAuth
insertionEncoding	commonName

Front end profile	Backend profile
nonFipsCiphers	sessReuse, sessTimeout
pushEncTrigger	SNIEnable
pushEncTriggerTimeout	ssl3
pushFlag	sslTriggerTimeout
quantumSize	strictCAChecks
redirectPortRewrite	tls1
sendCloseNotify	-
sessReuse, sessTimeout	-
SNIEnable	-
ssl3	-
sslRedirect	-
sslTriggerTimeout	-
strictCAChecks	-
tls1, tls11, tls12	-

* The clearTextPort parameter applies only to an SSL virtual server.

An error message appears if you try to set a parameter that is not part of the profile (for example, if you try to set the clientAuth parameter in a backend profile).

Some SSL parameters, such as CRL memory size, OCSP cache size, UndefAction Control, and UndefAction Data, are not part of any of the above profiles, because these parameters are independent of entities.

An SSL profile supports the following operations:

- Add—Creates an SSL profile on the NetScaler. Specify whether the profile is front end or backend. Front end is the default.
- Set—Modifies the settings of an existing profile.
- Unset—Sets the specified parameters to their default values. If you do not specify any parameters, an error message appears. If you unset a profile on an entity, the profile is unbound from the entity.
- Remove—Deletes a profile. A profile that is being used by any entity cannot be deleted. Clearing the configuration deletes all the entities. As a result, the profiles are also deleted.
- Show—Displays all the profiles that are available on the NetScaler . If a profile name is specified, the details of that profile are displayed. If an entity is specified, the profiles associated with that

entity are displayed.

Create an SSL profile by using the CLI:

- To add an SSL profile, type: `add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]`
- To modify an existing profile, type: `set ssl profile <name>`
- To unset an existing profile, type: `unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA]…`
- To unset an existing profile from an entity, type: `unset ssl vserver <vServerName> -sslProfile`
- To remove an existing profile, type: `rm ssl profile <name>`
- To display an existing profile, type: `sh ssl profile <name>`

Create an SSL profile by using the GUI:

Navigate to **System** > **Profiles**, select the SSL Profiles tab, and create an SSL profile.

Enable stricter control on client certificate validation

The NetScaler appliance accepts valid Intermediate-CA certificates if they are issued by a single Root-CA. That is, if only the Root-CA certificate is bound to the virtual server, and any intermediate certificate sent with the client certificate is validated by that Root-CA, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the ClientAuthUseBoundCAChain setting in the SSL profile bound to the virtual server. The handshake fails if the client certificate is not signed by one of the CA certificates bound to the virtual server.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, only clientcert1 is accepted by the NetScaler appliance

Enable stricter control on client certificate validation by using the CLI

At the command prompt, type:

```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

Enable stricter control on client certificate validation by using the GUI

1. Navigate to **System** > **Profiles**, select the **SSL Profiles** tab, and create an SSL profile, or select an existing profile.
2. Select **Enable Client Authentication using bound CA Chain**.

Certificate revocation lists

September 21, 2020

A certificate issued by a CA typically remains valid until its expiration date. However, in some circumstances, the CA may revoke the issued certificate before the expiration date (for example, when an owner's private key is compromised, a company's or individual's name changes, or the association between the subject and the CA changes).

A Certificate Revocation List (CRL) identifies invalid certificates by serial number and issuer.

Certificate authorities issue CRLs on a regular basis. You can configure the NetScaler appliance to use a CRL to block client requests that present invalid certificates.

If you already have a CRL file from a CA, add that to the NetScaler appliance. You can configure refresh options. You can also configure the NetScaler to sync the CRL file automatically at a specified interval, from either a web location or an LDAP location. The appliance supports CRLs in either the PEM or the DER file format. Be sure to specify the file format of the CRL file being added to the NetScaler appliance.

If you have used the NetScaler as a CA to create certificates that are used in SSL deployments, you can also create a CRL to revoke a particular certificate. This feature can be used, for example, to ensure that self-signed certificates that are created on the NetScaler are not used either in a production environment or beyond a particular date.

Note:

By default, CRLs are stored in the /var/netscaler/ssl directory on the NetScaler appliance.

Create a CRL on the ADC

Since you can use the NetScaler appliance to act as a certificate authority and create self-signed certificates, you can also revoke certificates that you have created and certificates whose CA certificate you own.

The appliance must revoke invalid certificates before creating a CRL for those certificates. The appliance stores the serial numbers of revoked certificates in an index file and updates the file each time it revokes a certificate. The index file is automatically created the first time a certificate is revoked.

Revoke a certificate or create a CRL by using the CLI

At the command prompt, type the following command:

```
create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <
    input_filename> | -genCRL <output_filename>)
<!--NeedCopy-->
```

Example:

```
create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -revoke Invalid-1

create ssl crl Cert-CA-1 Key-CA-1 File-Index-1 -genCRL CRL-1
<!--NeedCopy-->
```

Revoke a certificate or create a CRL by using the GUI

1. Navigate to Traffic Management > SSL and, in the Getting Started group, select CRL Management.
2. Enter the certificate details and, in the Choose Operation list, select Revoke Certificate or Generate CRL.

Add an existing CRL to the ADC

Before you configure the CRL on the NetScaler appliance, make sure that the CRL file is stored locally on the NetScaler appliance. In the case of an HA setup, the CRL file must be present on both NetScaler appliances, and the directory path to the file must be the same on both appliances.

Add a CRL on the NetScaler by using the CLI

At the command prompt, type the following commands to add a CRL on the NetScaler and verify the configuration:

```
add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

Example:

```
> add ssl crl crl-one /var/netscaler/ssl/CRL-one -inform PEM

Done

show ssl crl crl-one

            Name: crl-one   Status: Valid,  Days to expiration: 29

            CRL Path: /var/netscaler/ssl/CRL-one

            Format: PEM     CAcert: samplecertkey

            Refresh: DISABLED

            Version: 1

            Signature Algorithm: sha1WithRSAEncryption

            Issuer:  C=US,ST=California,L=Santa Clara,O=NetScaler Inc.,OU=SSL Acceleration,CN=www.ns.com/emailAddress=support@NetScaler appliance.com

            Last_update:Jun 15 10:53:53 2010 GMT

            Next_update:Jul 15 10:53:53 2010 GMT



    1)      Serial Number: 00

            Revocation Date:Jun 15 10:51:16 2010 GMT

     Done
<!--NeedCopy-->
```

Add a CRL on the NetScaler by using the GUI

Navigate to **Traffic Management** > **SSL** > **CRL**, and add a CRL.

Configure CRL refresh parameters

A CRL is generated and published by a Certificate Authority periodically or, in some cases, immediately after a particular certificate is revoked. Citrix recommends that you update CRLs on the NetScaler appliance regularly, for protection against clients trying to connect with certificates that are not valid.

The NetScaler appliance can refresh CRLs from a web location or an LDAP directory. When you specify refresh parameters and a web location or an LDAP server, the CRL does not have to be present on the local hard disk drive at the time you execute the command. The first refresh stores a copy on the local hard disk drive, in the path specified by the CRL File parameter. The default path for storing the CRL is /var/netscaler/ssl.

Note: In release 10.0 and later, the method for refreshing a CRL is not included by default. You must explicitly specify an HTTP or LDAP method. If you are upgrading from an earlier release to release 10.0 or later, you must add a method and run the command again.

Configure CRL autorefresh by using the CLI

At the command prompt, type the following commands to configure CRL auto refresh and verify the configuration the following commands to configure CRL auto refresh and verify the configuration:

```
set ssl crl <crlName> [-refresh ( ENABLED | DISABLED )] [-CAcert <string>] [-url <URL | -server <ip_addr|ipv6_addr>] [-method HTTP | ( LDAP [-baseDN <string>] [-bindDN <string>] [-scope ( Base | One )] [-password <string>] [-binary ( YES | NO )])] [-port <port>] [-interval <interval>]

show ssl crl [<crlName>]
<!--NeedCopy-->
```

Example:

```
set CRL crl1  -refresh enabled -method ldap -inform DER -CAcert ca1 -server 10.102.192.192 -port 389 -scope base -baseDN "cn=clnt_rsa4_multicert_der,ou=eng,o=ns,c=in" -time 00:01

set ssl crl crl1 -refresh enabled -method http -cacert ca1 -port 80 -time 00:10 -url http://10.102.192.192/crl/ca1.crl


sh crl

    1)         Name: crl1         Status: Valid,      Days to expiration: 355

```

```
                CRL Path: /var/netscaler/ssl/crl1

                Format: PEM      CAcert: ca1

                Refresh: ENABLED          Method: HTTP

                URL: http://10.102.192.192/crl/ca1.crl
         Port:80

                Refresh Time: 00:10

                Last Update: Successful, Date:Tue Jul  6 14:38:13 2010

Done
<!--NeedCopy-->
```

Configure CRL autorefresh using LDAP or HTTP by using the GUI

1. Navigate to Traffic Management > SSL > CRL.
2. Open a CRL, and select Enable CRL Auto Refresh.

Note: If the new CRL has been refreshed in the external repository before its actual update time as specified by the Last Update time field of the CRL, you should immediately refresh the CRL on the NetScaler appliance.

To view the last update time, select the CRL, and click Details.

Synchronize CRLs

The NetScaler appliance uses the most recently distributed CRL to prevent clients with revoked certificates from accessing secure resources.

If CRLs are updated often, the NetScaler appliance needs an automated mechanism to fetch the latest CRLs from the repository. You can configure the appliance to update CRLs automatically at a specified refresh interval.

The appliance maintains an internal list of CRLs that need to be updated at regular intervals. At these specified intervals, the appliance scans the list for CRLs that need to be updated, connects to the remote LDAP server or HTTP server, retrieves the latest CRLs, and then updates the local CRL list with the new CRLs.

Note: If CRL check is set to mandatory when the CA certificate is bound to the virtual server, and the initial CRL refresh fails, all client-authentication connections with the same issuer as the CRL are rejected as REVOKED until the CRL is successfully refreshed.

You can specify the interval at which the CRL refresh should be carried out. You can also specify the exact time.

Synchronize CRL autorefresh by using the CLI

At the command prompt, type the following command:

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <
    HH:MM>]
<!--NeedCopy-->
```

Example:

```
set ssl crl CRL-1 -refresh ENABLE -interval MONTHLY -days 10 -time
    12:00
<!--NeedCopy-->
```

Synchronize CRL refresh by using the GUI

1. Navigate to Traffic Management > SSL > CRL.
2. Open a CRL, select enable CRL Auto Refresh, and specify the interval.

Perform client authentication by using a certificate revocation list

If a certificate revocation list (CRL) is present on a NetScaler appliance, a CRL check is performed regardless of whether performing the CRL check is set to mandatory or optional.

The success or failure of a handshake depends on a combination of the following factors:

- Rule for CRL check
- Rule for client certificate check
- State of the CRL configured for the CA certificate

The following table lists the results of the possible combinations for a handshake involving a revoked certificate.

Table 1. Result of a Handshake with a Client Using a Revoked Certificate

Rule for CRL Check	Rule for Client Certificate Check	State of the CRL Configured for the CA certificate	Result of a Handshake with a Revoked Certificate
Optional	Optional	Missing	Success
Optional	Mandatory	Missing	Success

Rule for CRL Check	Rule for Client Certificate Check	State of the CRL Configured for the CA certificate	Result of a Handshake with a Revoked Certificate
Optional	Mandatory	Present	Failure
Mandatory	Optional	Missing	Success
Mandatory	Mandatory	Missing	Failure
Mandatory	Optional	Present	Success
Mandatory	Mandatory	Present	Failure
Optional/Mandatory	Optional	Expired	Success
Optional/Mandatory	Mandatory	Expired	Failure

> **Note:**
>
> - The CRL check is optional by default. To change from optional to mandatory or vice-versa, you must first unbind the certificate from the SSL virtual server, and then bind it again after changing the option.
> - In the output of the sh ssl vserver command, OCSP check: optional implies that a CRL check is also optional. The CRL check settings are displayed in the output of the sh ssl vserver command only if CRL check is set to mandatory. If CRL check is set to optional, the CRL check details do not appear.

To configure CRL check by using the CLI

At the command prompt, type the following command:

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck ( Mandatory | Optional ))]
sh ssl vserver
<!--NeedCopy-->
```

Example:

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
> sh ssl vs v1

Advanced SSL configuration for VServer v1:

DH: DISABLED
DH Private-Key Exponent Size Limit: DISABLED
```

```
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
OCSP Stapling: DISABLED
HSTS: DISABLED
HSTS IncludeSubDomains: NO
HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.1: ENABLED TLSv1.2: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES

ECC Curve: P_256, P_384, P_224, P_521

1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Configure CRL check by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open an SSL virtual server.
2. Click in the Certificates section.
3. Select a certificate and, in the OCSP and CRL Check list, select CRL Mandatory.

Monitor certificate status with OCSP

September 21, 2020

Online Certificate Status Protocol (OCSP) is an Internet protocol that is used to determine the status of a client SSL certificate. NetScaler appliances support OCSP as defined in RFC 2560. OCSP offers significant advantages over certificate revocation lists (CRLs) in terms of timely information. Up-to-date revocation status of a client certificate is especially useful in transactions involving large sums of money and high-value stock trades. It also uses fewer system and network resources. NetScaler implementation of OCSP includes request batching and response caching.

OCSP implementation

OCSP validation on a NetScaler appliance begins when the appliance receives a client certificate during an SSL handshake. To validate the certificate, the appliance creates an OCSP request and forwards it to the OCSP responder. To do so, the appliance uses a locally configured URL. The transaction is in a suspended state until the appliance evaluates the response from the server and determines whether to allow the transaction or reject it. If the response from the server is delayed beyond the configured time and no other responders are configured, the appliance will allow the transaction or display an error, depending on whether the OCSP check was set to optional or mandatory, respectively.

The appliance supports batching of OCSP requests and caching of OCSP responses to reduce the load on the OCSP responder and provide faster responses.

OCSP request batching

Each time the appliance receives a client certificate, it sends a request to the OCSP responder. To help avoid overloading the OCSP responder, the appliance can query the status of more than one client certificate in the same request. For this to work efficiently, a timeout needs to be defined so that processing of a single certificate is not inordinately delayed while waiting to form a batch.

OCSP response caching

Caching of responses received from the OCSP responder enables faster responses to the clients and reduces the load on the OCSP responder. Upon receiving the revocation status of a client certificate from the OCSP responder, the appliance caches the response locally for a predefined length of time. When a client certificate is received during an SSL handshake, the appliance first checks its local cache for an entry for this certificate. If an entry is found that is still valid (within the cache timeout limit), it is evaluated and the client certificate is accepted or rejected. If a certificate is not found, the appliance sends a request to the OCSP responder and stores the response in its local cache for a configured length of time.

OCSP responder configuration

Configuring OCSP involves adding an OCSP responder, binding the OCSP responder to a certification authority (CA) certificate, and binding the certificate to an SSL virtual server. If you need to bind a different certificate to an OCSP responder that has already been configured, you need to first unbind the responder and then bind the responder to a different certificate.

Add an OCSP responder by using the CLI

At the command prompt, type the following commands to configure OCSP and verify the configuration:

```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED ) [-cacheTimeout <positive_integer>]] [ -batchingDepth <positive_integer>][-batchingDelay <positive_integer>] [-resptimeout <positive_integer>] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][ -insertClientCert( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]
<!--NeedCopy-->
```

```
bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck ( Mandatory | Optional )]))
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Example:

```
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -cacheTimeout 30 -batchingDepth 8 -batchingDelay 100 -resptimeout 100 -responderCert responder_cert -producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
<!--NeedCopy-->
```

```
bind ssl certKey ca_cert -ocspResponder ocsp_responder1 -priority 1
<!--NeedCopy-->
```

```
bind ssl vserver vs1 -certkeyName ca_cert -CA -ocspCheck Mandatory
<!--NeedCopy-->
```

```
sh ocspResponder ocsp_responder1

    1)Name: ocsp_responder1

    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22

    Caching: Enabled        Timeout: 30 minutes

    Batching: 8 Timeout: 100 mS

    HTTP Request Timeout: 100mS

    Request Signing Certificate: sign_cert

    Response Verification: Full, Certificate: responder_cert

    ProducedAt Time Skew: 300 s

    Nonce Extension: Enabled

     Client Cert Insertion: Enabled

    Done
<!--NeedCopy-->
```

```
show certkey ca_cert

    Name: ca_cert     Status: Valid,   Days to expiration:8907

    Version: 3

    …

    1)  VServer name: vs1      CA Certificate

    1)  OCSP Responder name: ocsp_responder1     Priority: 1

Done
<!--NeedCopy-->
```

```
sh ssl vs vs1

    Advanced SSL configuration for VServer vs1:

    DH: DISABLED

    …

    1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory

    1) Cipher Name: DEFAULT

      Description: Predefined Cipher Alias

    Done
<!--NeedCopy-->
```

Modify an OCSP responder by using the CLI

You cannot modify the responder name. All other parameters can be changed using the set ssl ocspResponder command.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED) ] [-cacheTimeout <positive_integer>] [-batchingDepth <positive_integer>] [-batchingDelay <positive_integer>] [-resptimeout <positive_integer>] [ -responderCert <string> | -trustResponder][-producedAtTimeSkew <positive_integer>][-signingCert <string>] [-useNonce ( YES | NO )]

unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]

bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]

show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Configure an OCSP responder by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **OCSP Responder**, and configure an OCSP responder.
2. Navigate to **Traffic Management** > **SSL** > **Certificates**, select a certificate, and in the **Action** list, select **OCSP Bindings**. Bind an OCSP responder.
3. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, open a virtual server, and click in the Certificates section to bind a CA certificate.
4. Optionally, select **select OCSP Mandatory**.

OCSP stapling

September 21, 2020

The NetScaler implementation of CRL and OCSP reports the revocation status of client certificates only. To check the revocation status of a server certificate received during an SSL handshake, a client must send a request to a certificate authority.

For web sites with heavy traffic, many clients receive the same server certificate. If each client sent a query for the revocation status of the server certificate, the certificate authority would be inundated with OCSP requests to check the validity of the certificate.

OCSP stapling solution

To avoid unnecessary congestion, the NetScaler appliance now supports OCSP stapling. That is, the appliance can now send the revocation status of a server certificate to a client, at the time of the SSL handshake, after validating the certificate status from an OCSP responder. The revocation status of a server certificate is "stapled" to the response the appliance sends to the client as part of the SSL handshake. To use the OCSP stapling feature, you must enable it on an SSL virtual server and add an OCSP responder on the appliance.

Note:

- NetScaler appliances support OCSP stapling as defined in RFC 6066.
- OCSP stapling is supported only on the front-end of NetScaler appliances.

> **Important:**
>
> NetScaler support for OCSP stapling is limited to handshakes using TLS protocol version 1.0 or higher.

OCSP response caching of server certificates

During the SSL handshake, when a client requests the revocation status of the server certificate, the appliance first checks its local cache for an entry for this certificate. If a valid entry is found, it is evaluated and the server certificate and its status are presented to the client. If a revocation status entry is not found, the appliance sends a request for the revocation status of the server certificate to the OCSP responder. If it receives a response, it sends the certificate and the revocation status to the client. If the next update field is present in the OCSP response, the response is cached for the configured length of time (value specified in timeout field.)

OCSP stapling configuration

Configuring OCSP stapling involves enabling the feature and configuring OCSP. To configure OCSP, you must add an OCSP responder, bind the OCSP responder to a CA certificate, and bind the certificate to an SSL virtual server.

> **Note:**
>
> OCSP responders with only HTTP based URL are supported.

Enable OCSP stapling by using the CLI

At the command prompt, type:

```
set ssl vserver <name> -ocspstapling [ENABLED | DISABLED]
<!--NeedCopy-->
```

Example:

```
set ssl vserver vip1 -ocspStapling ENABLED
Done
sh ssl vserver vip1

     Advanced SSL configuration for VServer vip1:
     DH: DISABLED
     DH Private-Key Exponent Size Limit: DISABLED Ephemeral RSA: ENABLED Refresh Count: 0
     Session Reuse: ENABLED Timeout: 120 seconds
     Cipher Redirect: DISABLED
     SSLv2 Redirect: DISABLED
     ClearText Port: 0
     Client Auth: DISABLED
     SSL Redirect: DISABLED
     Non FIPS Ciphers: DISABLED
     SNI: ENABLED
     OCSP Stapling: ENABLED
     SSLv2: DISABLED SSLv3: DISABLED TLSv1.0: ENABLED  TLSv1.1: ENABLED TLSv1.2: ENABLED
     Push Encryption Trigger: Always
     Send Close-Notify: YES

     ECC Curve: P_256, P_384, P_224, P_521

    1) CertKey Name: server_certificate1 Server Certificate

    1) Cipher Name: DEFAULT
```

```
        Description: Default cipher list with encryption strength >= 128 bit
Done
<!--NeedCopy-->
```

Enable OCSP stapling by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Virtual Server**.
2. Open a virtual server and, in **SSL Parameters**, select **OCSP Stapling**.

OCSP configuration

An OCSP responder is added dynamically or manually to send OCSP stapling requests. An internal responder is dynamically added when you add a server certificate and its issuer certificate based on the OCSP URL in the server certificate. A manual OCSP responder is added from the CLI or GUI. To send an OCSP request for a server certificate, the NetScaler appliance selects an OCSP responder based on the priority assigned to it when binding it to an issuer certificate. If a responder fails to send an OCSP stapling request, the responder with the next highest priority is selected for sending the request. For example, if only one responder is manually configured and it fails and a dynamically bound responder exists, it is selected for sending the OCSP request.

If the OCSP URL is other than HTTP, an internal OCSP responder is not created.

> **Note**
>
> A manually added OCSP responder takes precedence over a dynamically added responder.

Difference between a manually created OCSP responder and an internally created OCSP responder

Manually created OCSP responder	Internally (dynamically) created OCSP responder
Created manually and explicitly bound to the issuer certificate with a priority.	Created and bound by default, while adding a server certificate and its issuer certificate (CA certificate). Name starts with "ns_internal_".
Priority between 1 and 127 is reserved for configured responder.	Priority is automatically assigned from 128 onwards.
URL and batching depth can be changed.	URL and batching depth cannot be changed.

Deleted directly.	Deleted only when you delete the server certificate or the CA certificate.
Can be bound to any CA certificate.	Bound by default to one CA certificate. Cannot be bound to any other CA certificate.
Saved in the configuration (ns.conf).	Add commands are not saved in the configuration; only set commands are saved.
If you bind three OCSP responders to the same issuer certificate with priorities 1, 2, and 3 respectively, and later unbind priority 2, the other priorities are not affected.	Three OCSP responders are automatically bound to an issuer certificate with priorities 128, 129, and 130 respectively. If you remove the server certificate that was used to create a responder bound with priority 129, then that responder is deleted. Also, the priority for the next responder (priority 130) is automatically changed to 129.

> **Example of request handling**:

1. Add a virtual server (VIP1).
2. Add issuer certificate (CA1) and bind it to VIP1.
3. Add three certificates S1, S2, and S3. Internal responders resp1, resp2, and resp3 respectively are created by default.
4. Bind S3 to VIP1.
5. A request comes to VIP1. Responder resp3 is selected.

To create an internal OCSP responder dynamically, the appliance needs the following:

- Certificate of the issuer of the server certificate (usually the CA certificate).
- Certificate-key pair of the server certificate. This certificate must contain the OCSP URL provided by the CA. The URL is used as the name of the dynamically added internal responder.

An internal OCSP responder has the same default values as a manually configured responder.

> **Note:**
>
> Caching is disabled by default on an internal responder. Use the "set ssl ocspResponder" command to enable caching.

Configure OCSP by using the CLI

At the command prompt, type the following commands to configure OCSP and verify the configuration:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>] [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

```
add ssl ocspResponder <name> -url <URL> [-cache ( ENABLED | DISABLED ) [-cacheTimeout <positive_integer>]] [-resptimeout <positive_integer>] [-responderCert <string> | -trustResponder] [-producedAtTimeSkew <positive_integer>][-signingCert <string>][-useNonce ( YES | NO )][-insertClientCert ( YES | NO )]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Parameters:

httpMethod:

HTTP method used to send OCSP requests. For requests, less than 255 bytes long, you can configure the HTTP GET method for queries to an OCSP server. If you specify the GET method but the length is greater than 255 bytes, the appliance uses the default method (POST).

Possible values: GET, POST

Default value: POST

ocspUrlResolveTimeout:

Time, in milliseconds, to wait for an OCSP URL resolution. After this time elapses, the responder with the next higher priority is selected. If all the responders fail, an error message appears or the connection is dropped, depending on the settings on the virtual server.

Minimum value: 100

Maximum value: 2000

Example:

```
add ssl certkey root_ca1 - cert root_cacert.pem
add ssl ocspResponder ocsp_responder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -cacheTimeout 30  -resptimeout 100 -responderCert responder_cert -producedAtTimeSkew 300 -signingCert sign_cert  -insertClientCert YES
bind ssl certKey root_ca1 -ocspResponder ocsp_responder1 -priority 1
sh ocspResponder ocsp_responder1
    1)Name: ocsp_responder1
    URL: http://www.myCA.org:80/ocsp/, IP: 192.128.22.22
    Caching: Enabled        Timeout: 30 minutes
    Batching: 8 Timeout: 100 mS
    HTTP Request Timeout: 100mS
    Request Signing Certificate: sign_cert
    Response Verification: Full, Certificate: responder_cert
    ProducedAt Time Skew: 300 s
    Nonce Extension: Enabled
     Client Cert Insertion: Enabled
    Done

show certkey root_ca1
    Name: root_ca1     Status: Valid,   Days to expiration:8907
    Version: 3
    …
    1)  OCSP Responder name: ocsp_responder1     Priority: 1
    Done
<!--NeedCopy-->
```

Modify OCSP by using the CLI

You cannot modify the name of an OCSP responder, but you can use the set ssl ocspResponder command to change any of the other parameters.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
set ssl ocspResponder <name> [-url <URL>] [-cache ( ENABLED | DISABLED) ] [-cacheTimeout <positive_integer>]  [-resptimeout <positive_integer>] [ -responderCert <string> | -trustResponder][-producedAtTimeSkew <positive_integer>][-signingCert <string>] [-useNonce ( YES | NO )]
<!--NeedCopy-->
```

```
unbind ssl certKey [<certkeyName>] [-ocspResponder <string>]
<!--NeedCopy-->
```

```
bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <
    positive_integer>]
<!--NeedCopy-->
```

```
show ssl ocspResponder [<name>]
<!--NeedCopy-->
```

Configure OCSP by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **OCSP Responder**, and configure an OCSP responder.
2. Navigate to **Traffic Management** > **SSL** > **Certificates**, select a certificate, and in the **Action** list, select **OCSP Bindings**. **Bind an OCSP responder**.
3. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, open a virtual server, and click in the Certificates section to bind a CA certificate.
4. Optionally, select select **OCSP Mandatory**.

> **Note:**
>
> The insert client certificate parameter in the "add ssl ocspResponder" and the "set ssl ocspResponder" commands is no longer valid. That is, the parameter is ignored during configuration.

Ciphers available on the NetScaler appliances

September 21, 2020

Your NetScaler appliance ships with a predefined set of cipher groups. To use ciphers that are not part of the DEFAULT cipher group, you have to explicitly bind them to an SSL virtual server. You can also create a user-defined cipher group to bind to the SSL virtual server.

For the list of ciphers supported on the different platforms, such as FIPS, VPX, external HSMs, MPX/SDX (N2), and MPX/SDX (N3), see https://docs.citrix.com/en-us/citrix-adc/current-release/ssl/ciphers-available-on-the-citrix-ADC-appliances.html

ECDHE ciphers

September 21, 2020

All NetScaler appliances support the ECDHE cipher group on the front end and the back end. On an SDX appliance, if an SSL chip is assigned to a VPX instance, the cipher support of an MPX appliance applies. Otherwise, the normal cipher support of a VPX instance applies.

For more information about the builds and platforms that support these ciphers, see Ciphers available on the NetScaler appliances.

ECDHE cipher suites use elliptical curve cryptography (ECC). Because of its smaller key size, ECC is especially useful in a mobile (wireless) environment or an interactive voice response environment, where every millisecond is important. Smaller key sizes save power, memory, bandwidth, and computational cost.

A NetScaler appliance supports the following ECC curves:

- P_256
- P_384
- P_224
- P_521

Note: If you upgrade from a build earlier than release 10.1 build 121.10, you must explicitly bind ECC curves to your existing SSL virtual servers or front end services. The curves are bound by default to any virtual servers or front end services that you create after the upgrade.

You can bind an ECC curve to SSL front-end entities only. By default all four curves are bound, in the following order: P_256, P_384,P_224,P_521. To change the order, you must first unbind all the curves, and then bind them in the desired order.

Unbind and bind ECC curves to an SSL virtual server by using the CLI

At the command prompt, type:

```
unbind ssl vserver <vServerName> -eccCurveName ALL

bind ssl vserver <vServerName> -eccCurveName <eccCurveName>
<!--NeedCopy-->
```

Example:

```
unbind ssl vs v1  - eccCurvename ALL

bind ssl vserver  v1 -eccCurveName P_224

sh ssl vserver v1

Advanced SSL configuration for VServer v1:

```

```
DH: DISABLED

Ephemeral RSA: ENABLED Refresh Count: 0

Session Reuse: ENABLED Timeout: 120 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SNI: DISABLED

SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED  TLSv1.1: DISABLED TLSv1.2: DISABLED

Push Encryption Trigger: Always

Send Close-Notify: YES

ECC Curve: P_224


1) Cipher Name: DEFAULT

Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Diffie-Hellman (DH) parameters generation and achieving PFS with DHE

September 21, 2020

The Diffie-Hellman (DH) key exchange is a way for two parties involved in an SSL transaction that have no prior knowledge of each other to agree upon a shared secret over an insecure channel. This secret can then be converted into cryptographic keying material for mainly symmetric key cipher algorithms that require such a key exchange.

This feature is disabled by default and should be specifically configured to support ciphers that use DH as the key exchange algorithm.

> **Note**:
>
> Generating 2048-bit DH params may take a long time (up to 30 minutes).

Generate DH parameters by using the CLI

At the command prompt, type the following command:

```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
<!--NeedCopy-->
```

Example:

```
create ssl dhparam Key-DH-1 512 -gen 2
<!--NeedCopy-->
```

Generate DH parameters by using the GUI

Navigate to **Traffic Management** > **SSL** and, in the **Tools** group, select **Create Diffie-Hellman (DH) key**, and **Configure SSL DH Param**.

> **Note:**
>
> For information about DH parameters, see Diffe-Hellman (DH) parameters.

Achieve perfect forward secrecy with DHE

Generating DH parameters is a CPU-intensive operation. In earlier releases, parameter generation, on a VPX appliance, took a long time because it was done in the software. Now, parameter generation is optimized by setting the `dhKeyExpSizeLimit` parameter. You can set this parameter for an SSL virtual server or an SSL profile and then bind the profile to a virtual server.

Additionally, to maintain perfect forward secrecy (PFS), DH count must ideally be zero. With this enhancement, you can generate DH params for each transaction (minimum DHcount is 0) without a significant drop in performance, because the operation is optimized. Earlier, the minimum DH count allowed was 500. That is, you could not regenerate the key for up to 500 transactions.

> **Important:**
>
> PFS is not supported on a VPX appliance.

Optimize DH parameters generation by using the CLI

At the command prompt, type commands 1 and 2, or type command 3:

```
1.  add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )] [-dhCount <positive_integer>] [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED)]
2.  set ssl vserver <vServerName> [-sslProfile <string>]
3.  set ssl vserver <vServerName> [-dh ( ENABLED | DISABLED) -dhFile <string>] [-dhCount <positive_integer>] [-dhKeyExpSizeLimit ( ENABLED | DISABLED )]
<!--NeedCopy-->
```

Optimize DH parameters generation by using the GUI

1. Navigate to **Traffic Management > Load Balancing > Virtual Servers**, and open a virtual server.
2. In the **SSL Parameters** section, select **Enable DH Key Expire Size Limit**.

Cipher redirection

September 21, 2020

During the SSL handshake, the SSL client (usually a web browser) announces the suite of ciphers that it supports, in the configured order of cipher preference. From that list, the SSL server then selects a cipher that matches its own list of configured ciphers.

If the ciphers announced by the client do not match those configured on the SSL server, the SSL handshake fails, and the failure is announced by a cryptic error message displayed in the browser. These messages rarely mention the exact cause of the error.

With cipher redirection, you can configure an SSL virtual server to deliver accurate, meaningful error messages when an SSL handshake fails. When SSL handshake fails, the NetScaler appliance redirects the user to a previously configured URL or, if no URL is configured, displays an internally generated error page.

Configure cipher redirection by using the CLI

At the command prompt, type the following commands to configure cipher redirection and verify the configuration:

```
set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED> -
    cipherURL < URL>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vs-ssl -cipherRedirect ENABLED -cipherURL http://
    redirectURl

Done

show ssl vserver vs-ssl

Advanced SSL configuration for VServer vs-ssl:

DH: DISABLED

Ephemeral RSA: ENABLED          Refresh Count: 1000

Session Reuse: ENABLED          Timeout: 600 seconds

Cipher Redirect: ENABLED        Redirect URL: http://redirectURl

SSLv2 Redirect: DISABLED

ClearText Port: 0

Client Auth: DISABLED

SSL Redirect: DISABLED

Non FIPS Ciphers: DISABLED

SNI: DISABLED

OCSP Stapling: DISABLED

HSTS: DISABLED

```

```
HSTS IncludeSubDomains: NO

HSTS Max-Age: 0

SSLv2: DISABLED SSLv3: ENABLED  TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1.2: ENABLED

    1)      CertKey Name: Auth-Cert-1       Server Certificate

    1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Configure cipher redirection by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Enable Cipher Redirect, and specify a redirect URL.

Leverage hardware and software to improve ECDHE cipher performance

September 21, 2020

> **Note:**
>
> This enhancement is applicable only to the following platforms:
>
> - MPX/SDX 11542
> - MPX/SDX 14000
> - MPX 22000, MPX 24000, and MPX 25000
> - MPX/SDX 14000 FIPS (from release 12.0-56.x)

Previously, ECDHE computation on a NetScaler appliance was performed only on the hardware (Cavium chips), which limited the number of SSL sessions at any given time. With this enhancement, some operations are also performed in the software. That is, processing is done both on the Cavium chips and on the CPU cores to improve ECDHE cipher performance.

The processing is first performed in software, up to the configured software crypto threshold. After this threshold is reached, the operations are offloaded to the hardware. Therefore, this hybrid model leverages both hardware and software to improve SSL performance. You can enable the hybrid model by setting the “softwareCryptoThreshold” parameter to suit your requirement. To disable the hybrid model, set this parameter to 0.

Benefits are greatest if the current CPU utilization is not too high, because the CPU threshold is not exclusive to ECDHE computation. For example, if the current workload on the NetScaler appliance consumes 50% of the CPU cycles, and the threshold is set to 80%, ECDHE computation can use an additional 30% of the cycles. After the configured software crypto threshold of 80% is reached, further ECDHE computation is offloaded to the hardware. In that case, actual CPU utilization might exceed 80%, because performing ECDHE computations in hardware consumes some CPU cycles.

Enable the hybrid model by using the CLI

At the command prompt, type:

```
set ssl parameter -softwareCryptoThreshold <positive_integer>

Synopsis:

softwareCryptoThreshold:

NetScaler CPU utilization threshold (as a percentage) beyond which crypto operations are not done in software. A value of zero implies that CPU is not utilized for doing crypto in software.

Default = 0

Min = 0

Max = 100
<!--NeedCopy-->
```

Example:

```
set ssl parameter - softwareCryptoThreshold 80
Done

show ssl parameter
Advanced SSL Parameters

SSL quantum size                    : 8 KB
Max CRL memory size                 : 256 MB
Strict CA checks                    : NO
Encryption trigger timeout          : 100 ms
Send Close-Notify                   : YES
```

```
Encryption trigger packet c       : 45
Deny SSL Renegotiation            : ALL
Subject/Issuer Name Insertion Format : Unicode
OCSP cache size                   : 10 MB
Push flag                         : 0x0 (Auto)
Strict Host Header check for SNI enabled SSL sessions : NO
PUSH encryption trigger timeout   : 1 ms
Crypto Device Disable Limit       : 0
Global undef action for control policies : CLIENTAUTH
Global undef action for data policies : NOOP
Default profile                   : DISABLED
Disable TLS 1.1/1.2 for SSL_BRIDGE secure monitors    : NO
Disable TLS 1.1/1.2 for dynamic and VPN services : NO
Software Crypto acceleration CPU Threshold : 80
Signature and Hash Algorithms supported by TLS1.2 : ALL
<!--NeedCopy-->
```

Enable the hybrid model by using the GUI

1. Navigate to **Traffic Management** > **SSL** > **Change advanced SSL settings**.
2. Enter a value for **Software Crypto Threshold (%)**.

ECDSA cipher suites support

September 21, 2020

ECDSA cipher suites use elliptical curve cryptography (ECC). Because of its smaller size, it is particularly helpful in environments where processing power, storage space, bandwidth, and power consumption are constrained.

The following NetScaler appliances now support the elliptical curve digital signature algorithm (ECDSA) cipher group:

- NetScaler MPX and SDX appliances with N3 chips
- NetScaler MPX 5900/8900
- NetScaler SDX 8900
- NetScaler VPX appliances

To support ECDSA cipher suites on a NetScaler SDX appliance, an SSL core must be assigned to the VPX instance.

When the ECDHE_ECDSA cipher group is used, the server's certificate must contain an ECDSA-capable public key.

Example:

```
sh ssl cipher ECDSA

1)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA       Priority : 1

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1   HexCode=0xc00a

2)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA       Priority : 2

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1   HexCode=0xc009

3)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 3

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384   HexCode=0xc024

4)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 4

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256   HexCode=0xc023

5)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384   Priority : 5

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD   HexCode=0xc02c

6)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256   Priority : 6

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD   HexCode=0xc02b

7)      Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA     Priority : 7

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1   HexCode=0xc008

8)      Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 8

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1   HexCode=0xc007
```

```

9)      Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305
     Priority : 9

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)
    Mac=AEAD   HexCode=0xcca9

Done
<!--NeedCopy-->
```

The following table lists the ECDSA ciphers that are supported on the NetScaler MPX and SDX appliances with N3 chips, MPX 5900, and MPX/SDX 8900 appliances.

Cipher Name	Priority	Description	Key Exchange Algorithm	Authenticat Algorithm	Encryption Algorithm (Key Size)	Message Authentication Code (MAC) Algorithm	HexCode
TLS1-ECDHE-ECDSA-AES128-SHA	1	SSLv3	ECC-DHE	ECDSA	AES(128)	SHA1	0xc009
TLS1-ECDHE-ECDSA-AES256-SHA	2	SSLv3	ECC-DHE	ECDSA	AES(256)	SHA1	0xc00a
TLS1.2-ECDHE-ECDSA-AES128-SHA256	3	TLSv1.2	ECC-DHE	ECDSA	AES(128)	SHA-256	0xc023
TLS1.2-ECDHE-ECDSA-AES256-SHA384	4	TLSv1.2	ECC-DHE	ECDSA	AES(256)	SHA-384	0xc024

TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256	5	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(128)	SHA-256	0xc02b
TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384	6	TLSv1.2	ECC-DHE	ECDSA	AES-GCM(256)	SHA-384	0xc02c
TLS1-ECDHE-ECDSA-RC4-SHA	7	SSLv3	ECC-DHE	ECDSA	RC4(128)	SHA1	0xc007
TLS1-ECDHE-ECDSA-DES-CBC3-SHA	8	SSLv3	ECC-DHE	ECDSA	3DES(168)	SHA1	0xc008
TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305	9	TLSv1.2	ECC-DHE	ECDSA	CHACHA20/POLY1305(256)	AEAD	0xcca9

> **Important**
>
> Use the **show ns hardware** command to find out if your appliance has N3 chips.

Example:

```
sh ns hardware
              Platform: NSMPX-22000 16*CPU+24*IX+12*E1K+2*E1K+4*CVM N3 2200100
              Manufactured on: 8/19/2013
              CPU: 2900MHZ
              Host Id: 1006665862
              Serial no: ENUK6298FT
              Encoded serial no: ENUK6298FT
Done
<!--NeedCopy-->
```

ECDSA/RSA cipher and certificate selection

You can bind both ECDSA and RSA server certificates at the same time to an SSL virtual server. When both ECDSA and RSA certificates are bound to the virtual server, it automatically selects the appropriate server certificate to present to the client. If the client cipher list includes RSA ciphers, but does not include ECDSA ciphers, the virtual server presents the RSA server certificate. If both ciphers are present in the client's list, then the server certificate presented depends on the cipher priority set on the virtual server. That is, if RSA has a higher priority, the RSA certificate is presented. If ECDSA has a higher priority, the ECDSA certificate is presented to the client.

Client authentication by using an ECDSA or an RSA certificate

For client authentication, the CA certificate bound to the virtual server can be ECDSA or RSA signed. The appliance supports a mixed certificate chain. For example, the following certificate chain is supported.

Client certificate (ECDSA) <-> CA certificate (RSA) <-> Intermediate certificate (RSA) <-> Root certificate (RSA)

> **Note**
>
> ECDSA certificates with only the following curves are supported:
>
> - prime256v1
> - secp384r1
> - secp521r1 (VPX only)
> - secp224r1 (VPX only)

Configure user-defined cipher groups on the ADC appliance

September 21, 2020

A cipher group is a set of cipher suites that you bind to an SSL virtual server, service, or service group on the NetScaler appliance. A cipher suite comprises a protocol, a key exchange (Kx) algorithm, an authentication (Au) algorithm, an encryption (Enc) algorithm, and a message authentication code (Mac) algorithm. Your appliance ships with a predefined set of cipher groups. When you create a SSL service or SSL service group, the ALL cipher group is automatically bound to it. However, when you create an SSL virtual server or a transparent SSL service, the DEFAULT cipher group is automatically bound to it. In addition, you can create a user-defined cipher group and bind it to an SSL virtual server, service, or service group.

Note: If your MPX appliance does not have any licenses, then only the EXPORT cipher is bound to your SSL virtual server, service, or service group.

To create a user-defined cipher group, first you create a cipher group and then you bind ciphers or cipher groups to this group. If you specify a cipher alias or a cipher group, all the ciphers in the cipher alias or group are added to the user-defined cipher group. You can also add individual ciphers (cipher suites) to a user-defined group. However, you cannot modify a predefined cipher group. Before removing a cipher group, unbind all the cipher suites in the group.

If you bind a cipher group to an SSL virtual server, service, or service group, the ciphers are appended to the existing ciphers that are bound to the entity. To bind a specific cipher group to the entity, you must first unbind the ciphers or cipher group that is bound to the entity and then bind the specific cipher group. For example, to bind only the AES cipher group to an SSL service, you perform the following steps:

1. Unbind the default cipher group ALL that is bound by default to the service when the service is created.

   ```
   unbind ssl service <service name> -cipherName ALL
   <!--NeedCopy-->
   ```

2. Bind the AES cipher group to the service

   ```
   bind ssl service <Service name> -cipherName AE
   <!--NeedCopy-->
   ```

If you want to bind the cipher group DES in addition to AES, at the command prompt, type:

```
bind ssl service <service name> -cipherName DES
<!--NeedCopy-->
```

Note: The free NetScaler virtual appliance supports only the DH cipher group.

Configure a user-defined cipher group by using the CLI

At the command prompt, type the following commands to add a cipher group, or to add ciphers to a previously created group, and verify the settings:

```
add ssl cipher <cipherGroupName>
bind ssl cipher <cipherGroupName> -cipherName <string>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

Example:

```
add ssl cipher test

Done

bind ssl cipher test -cipherName ECDHE

Done

sh ssl cipher test

1)      Cipher Name: TLS1-ECDHE-RSA-AES256-SHA  Priority : 1

Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA1   HexCode=0xc014

2)      Cipher Name: TLS1-ECDHE-RSA-AES128-SHA  Priority : 2

Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA1   HexCode=0xc013

3)      Cipher Name: TLS1.2-ECDHE-RSA-AES-256-SHA384    Priority : 3

Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(256)  Mac=SHA-384   HexCode=0xc028

4)      Cipher Name: TLS1.2-ECDHE-RSA-AES-128-SHA256    Priority : 4

Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES(128)  Mac=SHA-256   HexCode=0xc027

5)      Cipher Name: TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 Priority : 5

Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(256) Mac=AEAD   HexCode=0xc030

6)      Cipher Name: TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 Priority : 6

```

```
Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=AES-GCM(128) Mac=AEAD HexCode=0xc02f

7)      Cipher Name: TLS1-ECDHE-ECDSA-AES256-SHA        Priority : 7

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA1 HexCode=0xc00a

8)      Cipher Name: TLS1-ECDHE-ECDSA-AES128-SHA        Priority : 8

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA1 HexCode=0xc009

9)      Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-SHA384   Priority : 9

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(256)  Mac=SHA-384 HexCode=0xc024

10)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-SHA256   Priority : 10

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES(128)  Mac=SHA-256 HexCode=0xc023

11)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES256-GCM-SHA384 Priority : 11

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(256) Mac=AEAD HexCode=0xc02c

12)     Cipher Name: TLS1.2-ECDHE-ECDSA-AES128-GCM-SHA256 Priority : 12

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=AES-GCM(128) Mac=AEAD HexCode=0xc02b

13)     Cipher Name: TLS1-ECDHE-RSA-DES-CBC3-SHA        Priority : 13

Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=3DES(168) Mac=SHA1   HexCode=0xc012

14)     Cipher Name: TLS1-ECDHE-ECDSA-DES-CBC3-SHA      Priority : 14

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=3DES(168) Mac=SHA1 HexCode=0xc008

```

```
15)     Cipher Name: TLS1-ECDHE-RSA-RC4-SHA     Priority : 15

Description: SSLv3 Kx=ECC-DHE  Au=RSA  Enc=RC4(128)  Mac=SHA1   HexCode=0xc011

16)     Cipher Name: TLS1-ECDHE-ECDSA-RC4-SHA   Priority : 16

Description: SSLv3 Kx=ECC-DHE  Au=ECDSA Enc=RC4(128)  Mac=SHA1    HexCode=0xc007

17)     Cipher Name: TLS1.2-ECDHE-RSA-CHACHA20-POLY1305 Priority : 17

Description: TLSv1.2 Kx=ECC-DHE  Au=RSA  Enc=CHACHA20/POLY1305(256) Mac=AEAD   HexCode=0xcca8

18)     Cipher Name: TLS1.2-ECDHE-ECDSA-CHACHA20-POLY1305    Priority : 18

Description: TLSv1.2 Kx=ECC-DHE  Au=ECDSA Enc=CHACHA20/POLY1305(256)  Mac=AEAD   HexCode=0xcca9

Done
<!--NeedCopy-->
```

Unbind ciphers from a cipher group by using the CLI

At the command prompt, type the following commands to unbind ciphers from a user-defined cipher group, and verify the settings:

```
show ssl cipher <cipherGroupName>
unbind ssl cipher <cipherGroupName> -cipherName <string>
show ssl cipher <cipherGroupName>
<!--NeedCopy-->
```

Remove a cipher group by using the CLI

Note: You cannot remove a built-in cipher group. Before removing a user-defined cipher group, make sure that the cipher group is empty.

At the command prompt, type the following commands to remove a user-defined cipher group, and verify the configuration:

```
rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
```

```
show ssl cipher <cipherGroupName>

<!--NeedCopy-->
```

Example:

```
rm ssl cipher test Done

sh ssl cipher test ERROR: No such resource [cipherGroupName, test]
<!--NeedCopy-->
```

Configure a user-defined cipher group by using the GUI

Navigate to Traffic Management > SSL > Cipher Groups, and configure a cipher group.

To bind a cipher group to an SSL virtual server, service, or service group by using the CLI:

At the command prompt, type one of the following:

```
bind ssl vserver <vServerName> -cipherName <string>
bind ssl service <serviceName> -cipherName <string>
bind ssl serviceGroup <serviceGroupName> -cipherName <string>

<!--NeedCopy-->
```

Example:

```
bind ssl vserver ssl_vserver_test -cipherName test

Done

bind ssl service  nshttps -cipherName test

Done

bind ssl servicegroup  ssl_svc  -cipherName test

Done
<!--NeedCopy-->
```

To bind a cipher group to an SSL virtual server, service, or service group by using the GUI:

1. Navigate to Traffic Management > Load Balancing > Virtual Servers or navigate to Traffic Management > Load Balancing > Services or navigate to Traffic Management > Load Balancing > Service Groups, and open the virtual server, service, or service group.

2. In Advanced Settings, select SSL Ciphers, and bind a cipher group to the virtual server, service, or service group.

Server certificate support matrix on the ADC appliance

September 21, 2020

The NetScaler appliance supports the following server certificates.

Table 1: Support on Frontend (FE) and Backend (BE) Service

Server certificate/-Platform	MPX/SDX (N2 CHIPS) FE	MPX/SDX (N2 CHIPS) BE	MPX/SDX (N3 CHIPS) FE	MPX/SDX (N3 CHIPS) BE	VPX FE	VPX BE
MD5	Y	Y	Y	Y	Y	Y
SHA1	Y	Y	Y	Y	Y	Y
SHA224	Y	Y	Y	Y	Y	Y
SHA256	Y	Y	Y	Y	Y	Y
SHA384	Y	Y	Y	Y	Y	Y
SHA512	Y	Y	Y	Y	Y	Y
RSA Key	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024, 2048, 3072, and 4096 bits	1024 and 2048 bits
DH Key	1024 and 2048 bits	1024 and 2048 bits	1024 and 2048 bits	1024 and 2048 bits	1024 and 2048 bits	1024 and 2048 bits

Server certificate/Platform	MPX 9700/10500/12500/15500 FIPS with FW 2.2 FE	MPX 9700/10500/12500/15500 FIPS with FW 2.2 BE	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
MD5	Y	Y	Y	Y
SHA1	Y	Y	Y	Y
SHA224	Y	Y	Y	Y
SHA256	Y	Y	Y	Y
SHA384	Y	Y	Y	Y

Server certificate/Platform	MPX 9700/10500/12500/15500 FIPS with FW 2.2 FE	MPX 9700/10500/12500/15500 FIPS with FW 2.2 BE	MPX/SDX 14030/14060/14080 FIPS FE	MPX/SDX 14030/14060/14080 FIPS BE
SHA512	Y	Y	Y	Y
RSA Key	2048 bits	2048 bits	2048 and 3072 bits	2048 and 3072 bits
DH Key	N	N	N	N

> **Note:**
>
> - In release 11.1 and earlier, a NetScaler appliance supports the following "signature algorithms" extensions in the back end client hello message: RSA-MD5, RSA-SHA1, and RSA-SHA256.
>   Because SHA 384 and SHA 512 signature algortihms extensions are not supported by the NetScaler appliance, some servers, such as Windows IIS servers, reset the connection.
> - Starting release 12.0, a NetScaler appliance supports all the signature_algorithms extensions.

Client authentication

September 21, 2020

In a typical SSL transaction, the client that is connecting to a server over a secure connection checks the validity of the server by checking the server's certificate before initiating the SSL transaction. In some cases, however, you might want to configure the server to authenticate the client that is connecting to it.

With client authentication enabled on an SSL virtual server, the NetScaler appliance asks for the client certificate during the SSL handshake. The appliance checks the certificate presented by the client for normal constraints, such as the issuer signature and expiration date.

> **Note:**
>
> For the appliance to verify issuer signatures, the certificate of the CA that issued the client certificate must be installed on the appliance and bound to the virtual server that the client is transacting with.

If the certificate is valid, the appliance allows the client to access all secure resources. But if the certificate is invalid, the appliance drops the client request during the SSL handshake.

The appliance verifies the client certificate by first forming a chain of certificates, starting with the client certificate and ending with the root CA certificate for the client (for example, VeriSign). The root CA certificate may contain one or more intermediate CA certificates (if the client certificate is not directly issued by the root CA).

Before you enable client authentication on the NetScaler appliance, make sure that a valid client certificate is installed on the client. Then, enable client authentication for the virtual server that will handle the transactions. Finally, bind the certificate of the CA that issued the client certificate to the virtual server on the appliance .

> **Note:**
>
> A NetScaler MPX appliance supports a certificate-key pair size from 512 to 4096 bits. The certificate must be signed by using one of the following hash algorithms:

- MD5
- SHA-1
- SHA-224
- SHA-256
- SHA-384
- SHA-512

On an SDX appliance, if an SSL chip is assigned to a VPX instance, the certificate-key pair size support of an MPX appliance applies. Otherwise, the normal certificate-key pair size support of a VPX instance applies.

A NetScaler virtual appliance (VPX instance) supports certificates of at least 512 bits, up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate
- 4096-bit certificate on the physical server
- 2048-bit client certificate (if client authentication is enabled on the virtual server)

Provide the client certificate

Before you configure client authentication, a valid client certificate must installed on the client. A client certificate includes details about the specific client system that will create secure sessions with the NetScaler appliance. Each client certificate is unique and should be used by only one client system.

Whether you obtain the client certificate from a CA, use an existing client certificate, or generate a client certificate on the NetScaler appliance, you must convert the certificate to the correct format. On the NetScaler appliance, certificates are stored in either the PEM or DER format and must be converted to PKCS#12 format before they are installed on the client system. After converting the certificate and transferring it to the client, system, make sure that it is installed on that system and configured for the client application that will be part of the SSL transactions (for example, the web browser).

For instructions on how to convert a certificate from PEM or DER format to PKCS#12 format, see Import and convert SSL files.

For instructions on how to generate a client certificate, see Create a certificate.

Enable client-certificate based authentication

By default, client authentication is disabled on the NetScaler appliance, and all SSL transactions proceed without authenticating the client. You can configure client authentication to be either optional or mandatory as part of the SSL handshake.

If client authentication is optional, the appliance requests the client certificate but proceeds with the SSL transaction even if the client presents an invalid certificate. If client authentication is mandatory, the appliance terminates the SSL handshake if the SSL client does not provide a valid certificate.

Caution: Citrix recommends that you define proper access control policies before changing client-certificate-based authentication check to optional.

> **Note:**
>
> Client authentication is configured for individual SSL virtual servers, not globally.

Enable client-certificate based authentication by using the CLI

At the command prompt, type the following commands to enable the client-certificate-based authentication and verify the configuration:

```
set ssl vserver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-clientCert (MANDATORY | OPTIONAL)]
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
set ssl vserver vssl -clientAuth ENABLED -clientCert Mandatory
Done
show ssl vserver vssl

Advanced SSL configuration for VServer vssl:
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
```

```
Cipher Redirect: DISABLED
SSLv2 Redirect: DISABLED
ClearText Port: 0
Client Auth: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        OCSP Stapling: DISABLED
        HSTS: DISABLED
        HSTS IncludeSubDomains: NO
        HSTS Max-Age: 0
SSLv2: DISABLED SSLv3: ENABLED TLSv1.0: ENABLED TLSv1.2: ENABLED TLSv1
    .2: ENABLED

1) CertKey Name: sslckey Server Certificate

1) Policy Name: client_cert_policy Priority: 0

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
<!--NeedCopy-->
```

Enable client-certificate based authentication by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Client Authentication, and in the Client Certificate list, select Mandatory.

> **Note:**
>
> If client authentication is set to mandatory and if the client certificate contains policy extensions, certificate validation fails. From release 12.0-56.x, you can set a parameter in the front-end SSL profile to skip this check. The parameter is disabled by default. That is, the check is performed by default.

Skip the policy extension check during client authentication by using the CLI

At the command prompt, type:

```
set ssl profile ns_default_ssl_profile_frontend -clientauth ENABLED -
    skipClientCertPolicyCheck ENABLED

Parameter
```

```

skipClientCertPolicyCheck

                Control policy extension check, if present inside the
    X509 certificate chain. Applicable only if client authentication is
    enabled and client certificate is set to mandatory. Possible values
    functions as follows:

- ENABLED: Skip the policy check during client authentication.

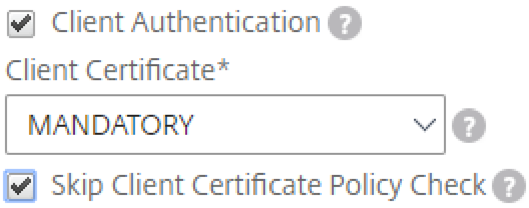
- DISABLED: Perform policy check during client authentication.

Possible values: ENABLED, DISABLED

Default: DISABLED
<!--NeedCopy-->
```

Skip the policy extension check during client authentication by using the GUI

1. Navigate to **System** > **Profiles** > **SSL Profiles**.
2. Create a new front-end profile or edit an existing front-end profile.
3. Verify that client authentication is enabled and client certificate is set to mandatory.
4. Select **Skip Client Certificate Policy Check**.



☑ Client Authentication

Client Certificate*: MANDATORY

☑ Skip Client Certificate Policy Check

Bind CA certificates to the virtual server

A CA whose certificate is present on the NetScaler appliance must issue the client certificate used for client authentication. You must bind this certificate to the NetScaler virtual server that will carry out client authentication.

You must bind the CA certificate to the SSL virtual server in such a way that the appliance can form a complete certificate chain when it verifies the client certificate. Otherwise, certificate chain formation fails and the client is denied access even if its certificate is valid.

You can bind CA certificates to the SSL virtual server in any order. The appliance forms the proper order during client certificate verification.

For example, if the client presents a certificate issued by **CA_A**, where **CA_A** is an intermediate CA whose certificate is issued by **CA_B**, whose certificate is in turn issued by a trusted root CA, **Root_CA**, a chain of certificates that contain all three of these certificates must be bound to the virtual server on the NetScaler appliance.

For instructions on binding one or more certificates to the virtual server, see Bind the certificate-key pair to the SSL virtual server.

For instructions on creating a chain of certificates, see Create a chain of certificates.

Stricter control on client certificate validation

The NetScaler appliance accepts valid Intermediate-CA certificates if they are issued by a single Root-CA. That is, if only the Root-CA certificate is bound to the virtual server, and any intermediate certificate sent with the client certificate is validated by that Root-CA, the appliance trusts the certificate chain and the handshake is successful.

However, if a client sends a chain of certificates in the handshake, none of the intermediate certificates can be validated by using a CRL or OCSP responder unless that certificate is bound to the SSL virtual server. Therefore, even if one of the intermediate certificates is revoked, the handshake is successful. As part of the handshake, the SSL virtual server sends the list of CA certificates that are bound to it. For stricter control, you can configure the SSL virtual server to accept only a certificate that is signed by one of the CA certificates bound to that virtual server. To do so, you must enable the ClientAuthUseBoundCAChain setting in the SSL profile bound to the virtual server. The handshake fails if the client certificate is not signed by one of the CA certificates bound to the virtual server.

For example, say two client certificates, clientcert1 and clientcert2, are signed by the intermediate certificates Int-CA-A and Int-CA-B, respectively. The intermediate certificates are signed by the root certificate Root-CA. Int-CA-A and Root-CA are bound to the SSL virtual server. In the default case (ClientAuthUseBoundCAChain disabled), both clientcert1 and clientcert2 are accepted. However, if ClientAuthUseBoundCAChain is enabled, only clientcert1 is accepted by the NetScaler appliance

Enable stricter control on client certificate validation by using the CLI

At the command prompt, type:

```
set ssl profile <name> -ClientAuthUseBoundCAChain Enabled
<!--NeedCopy-->
```

Enable stricter control on client certificate validation by using the GUI

1. Navigate to **System** > **Profiles**, select the **SSL Profiles** tab, and create an SSL profile, or select an existing profile.

2. Select **Enable Client Authentication using bound CA Chain**.

Server authentication

September 21, 2020

Since the NetScaler appliance performs SSL offload and acceleration on behalf of a web server, the appliance does not usually authenticate the Web server's certificate. However, you can authenticate the server in deployments that require end-to-end SSL encryption.

In such a situation, the appliance becomes the SSL client, carries out a secure transaction with the SSL server, verifies that a CA whose certificate is bound to the SSL service has signed the server certificate, and checks the validity of the server certificate.

To authenticate the server, you must first enable server authentication and then bind the certificate of the CA that signed the server's certificate to the SSL service on the NetScaler appliance. When binding the certificate, you must specify the bind as CA option.

Enable (or disable) server certificate authentication

You can use the CLI and the GUI to enable and disable server certificate authentication.

Enable (or disable) server certificate authentication using the CLI

At the command prompt, type the following commands to enable server certificate authentication and verify the configuration:

```
set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )
show ssl service <serviceName>
<!--NeedCopy-->
```

Example:

```
set ssl service ssl-service-1 -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-service-1:`

            DH: DISABLED

            Ephemeral RSA: DISABLED

            Session Reuse: ENABLED          Timeout: 300 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            Server Auth: ENABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

    1)      Cipher Name: ALL

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Enable (or disable) server certificate authentication by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services**, and open an SSL service.
2. In the **SSL Parameters** section, select **Enable Server Authentication**, and specify a Common Name.
3. In **Advanced Settings**, select **Certificates**, and bind a CA certificate to the service.

Bind the CA certificate to the service by using the CLI

At the command prompt, type the following commands to bind the CA certificate to the service and verify the configuration:

```
bind ssl service <serviceName> -certkeyName <string> -CA

show ssl service <serviceName>
<!--NeedCopy-->
```

Example:

```
bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
<!--NeedCopy-->
```

```
show ssl service ssl-service-1

            Advanced SSL configuration for Back-end SSL Service ssl-service-1:

            DH: DISABLED

            Ephemeral RSA: DISABLED

            Session Reuse: ENABLED          Timeout: 300 seconds

            Cipher Redirect: DISABLED

            SSLv2 Redirect: DISABLED

            Server Auth: ENABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

    1)      CertKey Name: samplecertkey     CA Certificate CRLCheck: Optional

    1)      Cipher Name: ALL

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Configure a common name for server certificate authentication

In end-to-end encryption with server authentication enabled, you can include a common name in the configuration of an SSL service or service group. The name that you specify is compared to the common name in the server certificate during an SSL handshake. If the two names match, the handshake is successful.

If the common names do not match, the common name specified for the service or service group is compared to values in the subject alternative name (SAN) field in the certificate. If it matches one of those values, the handshake is successful. This configuration is especially useful if there are, for example, two servers behind a firewall and one of the servers spoofs the identity of the other. If the common name is not checked, a certificate presented by either server is accepted if the IP address matches.

> **Note:**
>
> Only domain name, URL, and email ID DNS entries in the SAN field are compared.

Configure common-name verification for an SSL service or service group by using the CLI

At the command prompt, type the following commands to specify server authentication with common-name verification and verify the configuration:

1. To configure common name in a service, type:

```
set ssl service <serviceName> -commonName <string> -serverAuth
    ENABLED
show ssl service <serviceName>
<!--NeedCopy-->
```

2. To configure common name in a service group, type:

```
set ssl serviceGroup <serviceGroupName> -commonName <string> -
    serverAuth ENABLED
show ssl serviceGroup <serviceGroupName>
<!--NeedCopy-->
```

Example:

```
set ssl service svc1 -commonName xyz.com -serverAuth ENABLED
<!--NeedCopy-->
```

```
show ssl service svc

     Advanced SSL configuration for Back-end SSL Service svc1:

     DH: DISABLED

     Ephemeral RSA: DISABLED

     Session Reuse: ENABLED Timeout: 300 seconds

     Cipher Redirect: DISABLED

     SSLv2 Redirect: DISABLED

     Server Auth: ENABLED Common Name: www.xyz.com

     SSL Redirect: DISABLED

     Non FIPS Ciphers: DISABLED

     SNI: DISABLED

     SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED


    1) CertKey Name: cacert CA Certificate OCSPCheck: Optional

    1) Cipher Name: ALL

     Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Configure common-name verification for an SSL service or service group by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Services** or Navigate to **Traffic Management** > **Load Balancing** > **Service Groups**, and open a service or service group.
2. In the **SSL Parameters** section, select **Enable Server Authentication**, and specify a common name.

SSL actions and policies

September 21, 2020

An SSL policy evaluates incoming traffic and applies a predefined action to requests that match a rule (expression). You have to configure the actions before creating the policies, so that you can specify an action when you create a policy. To put a policy into effect, you must either bind it to a virtual server on the appliance, so that it applies only to traffic flowing through that virtual server, or bind it globally, so that it applies to all traffic flowing through the appliance.

SSL actions define SSL settings that you can apply to the selected requests. You associate an action with one or more policies. Data in client connection requests or responses is compared to a rule specified in the policy, and the action is applied to connections that match the rule (expression).

You can configure classic policies with classic expressions and default syntax policies with default syntax expressions for SSL.

> **Note:**
>
> Users who are not experienced in configuring policies at the CLI usually find using the configuration utility to be considerably easier.

You can associate a user-defined action or a built-in action to a default syntax policy. Classic policies allow only user-defined actions. In default syntax policy, you can also group policies under a policy label, in which case they are applied only when invoked from another policy.

Common uses of SSL actions and policies include per-directory client authentication, support for Outlook web access, and SSL-based header insertions. SSL-based header insertions contain SSL settings required by a server whose SSL processing has been offloaded to the NetScaler appliance.

SSL policies

September 21, 2020

Policies on the NetScaler appliance help identify specific connections that you want to process. The processing is based on the actions that are configured for that particular policy. Once you create the policy and configure an action for it, you must either bind it to a virtual server on the appliance, so that it applies only to traffic flowing through that virtual server, or bind it globally, so that it applies to all traffic flowing through any virtual server configured on the NetScaler appliance.

The NetScaler appliance SSL feature supports default syntax (advanced) policies . For a complete description of default syntax expressions, how they work, and how to configure them manually, see Policies and Expressions.

> **Note:**
>
> Users who are not experienced in configuring policies at the CLI will usually find using the configuration utility considerably easier.

SSL policies require that you create an action before creating a policy, so that you can specify the actions when you create the policies.
In SSL default syntax policies, you can also use the built-in actions. For more information about built-in actions, see SSL built-in actions and user-defined actions.

SSL default syntax policies

An SSL default syntax policy, also known as advanced policy, defines a control or a data action to be performed on requests. SSL policies can therefore be categorized as control policies and data policies:

- **Control policy**. A control policy uses a control action, such as forcing client authentication. Note: In release 10.5 or later, deny SSL renegotiation (denySSLReneg) is set, by default, to ALL. However, control policies, such as CLIENTAUTH, trigger a renegotiation handshake. If you use such policies, you must set denySSLReneg to NO.
- **Data policy**. A data policy uses a data action, such as inserting some data into the request.

The essential components of a policy are an expression and an action. The expression identifies the requests on which the action is to be performed.

You can configure a default syntax policy with a built-in action or a user-defined action. You can configure a policy with a built-in action without creating a separate action. However, to configure a policy with a user-defined action, first configure the action and then configure the policy.

You can specify an additional action, called an UNDEF action, to be performed in the event that applying the expression to a request has an undefined result.

SSL policy configuration

You can configure an SSL default syntax policy by using the CLI and the GUI.

Configure an SSL policy by using the CLI

At the command prompt, type:

```
add ssl policy <name> -rule <expression> -Action <string> [-undefAction <string>] [-comment <string>]
<!--NeedCopy-->
```

Configure an SSL policy by using the GUI

Navigate to **Traffic Management** > **SSL** > **Policies** and, on the **Polices** tab, click **Add**.

SSL built-in actions and user-defined actions

September 21, 2020

Unless you need only the built-in actions in your policies, you must create the actions before creating the policies. Then, you can specify the actions when you create the policies. The built-in actions are of two types, control actions and data actions. You use control actions in control policies, and data actions in data policies.

The built-in control actions are:

- CLIENTAUTH—Perform client certificate authentication.
- NOCLIENTAUTH—Do not perform client certificate authentication.

The built-in data actions are:

- RESET—Close the connection by sending an RST packet to the client.
- DROP—Drop all packets from the client. The connection remains open until the client closes it.
- NOOP—Forward the packet without performing any operation on it.

You can create user-defined data actions. For example, if you enable client authentication, you can create an SSL action to insert client-certificate data into the request header before forwarding the request to the web server.

If a policy evaluation results in an undefined state, an UNDEF action is performed. For either a data policy or a control policy, you can specify RESET, DROP, or NOOP as the UNDEF action. For a control policy, you also have the option of specifying CLIENTAUTH or NOCLIENTAUTH.

Examples of built-in actions in a policy

In the following example, if the client sends a cipher other than an EXPORT category cipher, the NetScaler appliance requests client authentication. The client has to provide a valid certificate for a successful transaction.

```
add ssl policy pol1 -rule CLIENT.SSL.CIPHER_EXPORTABLE.NOT -reqAction CLIENTAUTH
<!--NeedCopy-->
```

The following examples assume that client authentication is enabled.

If the version in the certificate provided by the user matches the version in the policy, no action is taken and the packet is forwarded:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -reqAction NOOP
<!--NeedCopy-->
```

If the version in the certificate provided by the user matches the version in the policy, the connection is dropped:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction DROP
<!--NeedCopy-->
```

If the version in the certificate provided by the user matches the version in the policy, the connection is reset:

```
add ssl policy pol1 -rule CLIENT.SSL.CLIENT_CERT.VERSION.EQ(2) -
    reqAction RESET
<!--NeedCopy-->
```

User-defined SSL actions

In addition to built-in actions, you can also configure other SSL actions depending on your deployment. These actions are called user-defined actions.

Configure a user-defined SSL action by using the CLI

At the command prompt, type the following commands to configure an action and verify the configuration:

```
add SSL action <name> -clientAuth(DOCLIENTAUTH | NOCLIENTAUTH) -
    clientCert (ENABLED | DISABLED) certHeader <string> -clientHeader <
    string> -clientCertSerialNumber (ENABLED | DISABLED) -
    certSerialHeader <string> -clientCertSubject (ENABLED | DISABLED) -
    certSubjectHeader <string> -clientCertHash (ENABLED | DISABLED) -
    certHashHeader <string> -clientCertIssuer (ENABLED | DISABLED) -
    certIssuerHeader <string> -sessionID (ENABLED | DISABLED) -
    sessionIDheader <string> -cipher (ENABLED | DISABLED) -cipherHeader
    <string> -clientCertNotBefore (ENABLED | DISABLED) -
    certNotBeforeHeader <string> -clientCertNotAfter (ENABLED | DISABLED
    ) -certNotAfterHeader <string> -OWASupport (ENABLED | DISABLED)
<!--NeedCopy-->
```

```
show ssl action [<name>]
<!--NeedCopy-->
```

Example:

```
add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X
    -Client-Cert"
<!--NeedCopy-->
```

```
show ssl action Action-SSL-ClientCert

1)       Name: Action-SSL-ClientCert

         Data Insertion Action:

         Cert Header: ENABLED            Cert Tag: X-Client-Cert

Done
<!--NeedCopy-->
```

Configure a user-defined SSL action by using the GUI

Navigate to **Traffic Management** > **SSL** > **Policies** and, on the **Actions** tab, click **Add**.

SSL policy binding

September 21, 2020

You can bind SSL policies globally or to an SSL virtual server. Globally bound policies are evaluated after all policies bound to services, virtual servers, or other NetScaler bind points are evaluated. If the incoming data matches any of the rules configured in the SSL policy, the policy is triggered and the action associated with it is carried out.

When binding an SSL policy to a virtual server, you must select from one of the following bind points:

- REQUEST (Default bind point. Policy evaluation is done in the HTTP layer after the SSL handshake is completed.)
- INTERCEPT_REQ (available in release 12.0 and later).

Similarly, when unbinding a policy from a virtual server, you must specify the bind point.

Bind an SSL policy globally by using the CLI

At the command prompt, type the following command to bind a global SSL policy and verify the configuration:

```
bind ssl global - policyName <string> [- priority <positive_integer>]
show ssl global
<!--NeedCopy-->
```

Example:

```
bind ssl global -policyName Policy-SSL-2 -priority 90

Done

sh ssl global

     1) Name: Policy-SSL-2 Priority: 90

     2) Name: Policy-SSL-1 Priority: 100

Done
<!--NeedCopy-->
```

Bind an SSL policy globally by using the GUI

1. Navigate to Traffic Management > SSL > Policies.
2. In the details pane, click Global Bindings.
3. In the Bind/Unbind SSL Policies to Global dialog box, click Insert Policy.
4. In the Policy Name drop-down list, select a policy.
5. Optionally, drag the entry to a new position in the policy bank to automatically update the priority level.
6. Click OK. A message appears in the status bar, stating that the policy has been bound successfully.

Bind or unbind an SSL policy to a virtual server by using the CLI

At the command prompt, type the following command to bind an SSL policy to a virtual server and verify the configuration:

```
bind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

unbind ssl vserver <vServerName> -policyName <string> -priority <
    positive_integer> -type <type>

<!--NeedCopy-->
```

Example:

```
bind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
unbind ssl vserver v1 -policyName pol1 -priority 1 -type CLIENTHELLO_REQ
<!--NeedCopy-->
```

```
show ssl vserver vs-server

Advanced SSL configuration for VServer vs-server:

DH: DISABLED

Ephemeral RSA: ENABLED          Refresh Count: 1000

Session Reuse: ENABLED          Timeout: 120 seconds

Cipher Redirect: DISABLED

SSLv2 Redirect: DISABLED

ClearText Port: 80

Client Auth: DISABLED

SSL Redirect: ENABLED

SSL-REDIRECT Port Rewrite: ENABLED

Non FIPS Ciphers: DISABLED

SSLv2: DISABLED SSLv3: ENABLED  TLSv1: ENABLED

1)      Policy Name: ssl-policy-1        Priority: 10

1)      Cipher Name: DEFAULT

            Description: Predefined Cipher Alias

Done
<!--NeedCopy-->
```

Bind an SSL policy to a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open an SSL virtual server.

2. In Advanced Settings, select SSL Policy, Click in the SSL Policy section to bind to the virtual server.
3. In the Policy Binding page, select an existing policy or add a new policy.
4. Specify priority and type (bind point) for the policy.
5. Click Bind.
6. Click Done.

SSL policy labels

September 21, 2020

Policy labels are holders for policies. A policy label helps in managing a group of policies, called a policy bank, which can be invoked from another policy. SSL policy labels can be control labels or data labels, depending on the type of policies that are included in the policy label. You can add only data policies in a data policy label and only control policies in a control policy label. To create the policy bank, you bind policies to the label and specify the order of evaluation of each policy relative to others in the bank of policies for the policy label. At the CLI, you enter two commands to create a policy label and bind policies to the policy label. In the configuration utility, you select options from a dialog box.

Create an SSL policy label and bind policies to the label by using the CLI

At the command prompt, type:

```
add ssl policylabel <labelName> -type ( CONTROL | DATA )

bind ssl policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>) ]
<!--NeedCopy-->
```

Example:

```
add ssl policylabel cpl1 -type CONTROL

add ssl policylabel dpl1 -type DATA

bind ssl policylabel cpl1 -policyName ctrlpol -priority 1

bind ssl policylabel dpl1 -policyName datapol -priority 1
<!--NeedCopy-->
```

Configure an SSL policy label and bind policies to the label by using the GUI

Navigate to **Traffic Management** > **SSL** > **Policy Labels**, and configure an SSL policy label.

Support for DTLS protocol

September 21, 2020

> **Note**
>
> DTLSv1.0 protocol is supported on Citrix MPX/SDX (N2 and N3 based) and VPX appliances. It is not supported on FIPS appliances, Intel Coleto SSL chip based appliances, and external HSMs.

The SSL and TLS protocols have traditionally been used to secure streaming traffic. Both of these protocols are based on TCP, which is very slow. In addition, TLS cannot handle lost or reordered packets.

UDP is the preferred protocol for audio and video applications, such as Lync, Skype, iTunes, YouTube, training videos, and flash. However, UDP is not secure or reliable. The DTLS protocol is designed to secure data over UDP and is used for applications such as media streaming, VOIP, and online gaming for communication. In DTLS, each handshake message is assigned a specific sequence number within that handshake. When a peer receives a handshake message, it can quickly determine whether that message is the next one expected. If it is, the peer processes the message. If not, the message is queued for handling after all the previous messages have been received.

You must create a DTLS virtual server and a service of type UDP. By default, a DTLS profile (nsdtls_default_profile) is bound to the virtual server. Optionally, you can create and bind a user-defined DTLS profile to the virtual server.

> **Note:**
>
> RC4 ciphers are not supported on a DTLS virtual server.

DTLS configuration

You can use the command line (CLI) or the configuration utility (GUI) to configure DTLS on your ADC appliance.

Create a DTLS configuration by using the CLI

At the command prompt, type:

```
add lb vserver <vserver_name> DTLS <IPAddress>  <port>
add service  <service_name>  <IPAddress> UDP 443
bind lb vserver  <vserver_name>  <udp_service_name>
```

```
<!--NeedCopy-->
```

The following steps are optional:

```
add dtlsProfile dtls1 -maxretryTime <positive_integer>
set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
<!--NeedCopy-->
```

Create a DTLS configuration by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Create a virtual server of type DTLS, and bind a UDP service to the virtual server.
3. A default DTLS profile is bound to the DTLS virtual server. To bind a different profile, in SSL Parameters, select a different DTLS profile. To create a new profile, click the plus (+) next to DTLS Profile.

Features not supported by a DTLS virtual server

The following options cannot be enabled on a DTLS virtual server:

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- Push encrypt trigger
- SSLv2Redirect
- SSLv2URL
- SNI
- Secure renegotiation

Parameters not used by a DTLS virtual server

The following SSL parameters, even if set, are ignored by a DTLS virtual server:

- Encryption trigger packet count
- PUSH encryption trigger timeout
- SSL quantum size
- Encryption trigger timeout
- Subject/Issuer Name Insertion Format

Features not supported by a DTLS service

The following options cannot be enabled on a DTLS service:

- SSLv2
- SSLv3
- TLSv1
- TLSv1.1
- TLSv1.2
- Push encrypt trigger
- SSLv2Redirect
- SSLv2URL
- SNI
- Secure renegotiation

Parameters not used by a DTLS service

The following SSL parameters, even if set, are ignored by a DTLS service:

- Encryption trigger packet count
- PUSH encryption trigger timeout
- SSL quantum size
- Encryption trigger timeout
- Subject/Issuer Name Insertion Format

> **Note:**
>
> SSL session reuse handshake fails on a DTLS service because session reuse is not currently supported on DTLS services.
>
> **Workaround:** Manually disable session reuse on a DTLS service. At the CLI, type:
>
> ```
> set ssl service <dtls-service-name> -sessReuse DISABLED
> <!--NeedCopy-->
> ```

DTLS profile

A DTLS profile with the default settings is automatically bound to a DTLS virtual server. However, you can create a new DTLS profile with specific settings to suit your requirement.

You must use a DTLS profile with a DTLS virtual server or a VPN DTLS virtual server. You cannot use an SSL profile with a DTLS virtual server.

Create a DTLS profile by using the CLI

```
add ssl dtlsProfile <name>

show ssl dtlsProfile<name>
<!--NeedCopy-->
```

Example:

```
add dtlsProfile dtls1 -helloVerifyRequest ENABLED -maxretryTime 4

Done

show dtlsProfile dtls1

    1)      Name: dtls1


            PMTU Discovery: DISABLED


            Max Record Size: 1460 bytes


            Max Retry Time: 4 sec


            Hello Verify Request: ENABLED


            Terminate Session: DISABLED


            Max Packet Count: 120 bytes

     Done
<!--NeedCopy-->
```

Create a DTLS profile by using the GUI

Navigate to **System** > **Profiles** > **DTLS Profiles** and configure a new profile.

Example for an end-to-end DTLS configuration

```
enable ns feature SSL LB

add server s1 198.51.100.2

en ns mode usnip

add service svc_dtls s1 DTLS 443

add lb vserver v1 DTLS 10.102.59.244 443

bind ssl vserver v1 -ciphername ALL

add ssl certkey servercert -cert servercert_aia_valid.pem -key serverkey_aia.pem

bind ssl vserver v1 -certkeyname servercert

bind lb vserver lb1 svc_dtls

sh lb vserver v1

                    v1 (10.102.59.244:4433) - DTLS     Type: ADDRESS

                    State: UP

                    Last state change was at Fri Apr 27 07:00:27 2018

                    Time since last state change: 0 days, 00:00:04.810

                    Effective State: UP

                    Client Idle Timeout: 120 sec

                    Down state flush: ENABLED

                    Disable Primary Vserver On Down : DISABLED

                    Appflow logging: ENABLED

                    No. of Bound Services :  1 (Total) 0 (Active)

                    Configured Method: LEASTCONNECTION

                    Current Method: Round Robin, Reason: A new service
```

```
       is bound         BackupMethod: ROUNDROBIN

                    Mode: IP

                    Persistence: NONE

                    L2Conn: OFF

                    Skip Persistency: None

                    Listen Policy: NONE

                    IcmpResponse: PASSIVE

                    RHIstate: PASSIVE

                    New Service Startup Request Rate: 0 PER_SECOND,
       Increment Interval: 0

                    Mac mode Retain Vlan: DISABLED

                    DBS_LB: DISABLED

                    Process Local: DISABLED

                    Traffic Domain: 0

                    TROFS Persistence honored: ENABLED

                    Retain Connections on Cluster: NO


    1) svc_dtls (10.102.59.190: 4433) - DTLS State: UP  Weight: 1

Done


sh ssl vserver v1

                    Advanced SSL configuration for VServer v1:




                    DH: DISABLED
```

```

                DH Private-Key Exponent Size Limit: DISABLED        Ephemeral RSA: ENABLED                              Refresh Count: 0

                Session Reuse: ENABLED                 Timeout: 1800 seconds

                Cipher Redirect: DISABLED

                ClearText Port: 0




                Client Auth: DISABLED

                SSL Redirect: DISABLED

                Non FIPS Ciphers: DISABLED

                SNI: DISABLED

                OCSP Stapling: DISABLED

                HSTS: DISABLED

                HSTS IncludeSubDomains: NO

                HSTS Max-Age: 0

                DTLSv1: ENABLED

                Send Close-Notify: YES

                Strict Sig-Digest Check: DISABLED

                Zero RTT Early Data: DISABLED

                DHE Key Exchange With PSK: NO

                Tickets Per Authentication Context: 1

                DTLS profile name: nsdtls_default_profile

```

```



                    ECC Curve: P_256, P_384, P_224, P_521


    1)            CertKey Name: servercert               Server Certificate


    1)            Cipher Name: DEFAULT

                    Description: Default cipher list with encryption strength >= 128bit


    2)            Cipher Name: ALL

                    Description: All ciphers supported by NetScaler, excluding NULL ciphers

     Done

sh service svc_dtls

                    svc_dtls (10.102.59.190:4433) - DTLS

                    State: UP

                    Last state change was at Fri Apr 27 07:00:26 2018

                    Time since last state change: 0 days, 00:00:22.790

                    Server Name: s1

                    Server ID : None                 Monitor Threshold : 0

                    Max Conn: 0        Max Req: 0           Max Bandwidth: 0 kbits

                    Use Source IP: NO

                    Client Keepalive(CKA): NO

                    Access Down Service: NO

                    TCP Buffering(TCPB): NO

                    HTTP Compression(CMP): NO

                    Idle timeout: Client: 120 sec           Server: 120 sec

                    Client IP: DISABLED

                    Cacheable: NO

                    SC: OFF

                    SP: OFF

                    Down state flush: ENABLED

                    Monitor Connection Close : NONE

                    Appflow logging: ENABLED

                    Process Local: DISABLED

                    Traffic Domain: 0



    1)            Monitor Name: ping-default

                                    State: UP               Weight: 1       Passive: 0

                                    Probes: 5              Failed [Total: 0 Current: 0]

                                    Last response: Success - ICMP echo reply received.

                                    Response Time: 2.77 millisec

     Done

sh ssl service svc_dtls
```

```

            Advanced SSL configuration for Back-end SSL Service svc_dtls:




            DH: DISABLED

            DH Private-Key Exponent Size Limit: DISABLED        Ephemeral RSA: DISABLED

            Session Reuse: ENABLED                 Timeout: 1800 seconds

            Cipher Redirect: DISABLED

            ClearText Port: 0




            Server Auth: DISABLED

            SSL Redirect: DISABLED

            Non FIPS Ciphers: DISABLED

            SNI: DISABLED

            OCSP Stapling: DISABLED

            DTLSv1: ENABLED

            Send Close-Notify: YES

            Strict Sig-Digest Check: DISABLED

            Zero RTT Early Data: ???

            DHE Key Exchange With PSK: ???

            Tickets Per Authentication Context: ???


```

```

                    DTLS profile name: nsdtls_default_profile


                    ECC Curve: P_256, P_384, P_224, P_521



    1)            Cipher Name: DEFAULT_BACKEND

                    Description: Default cipher list for Backend SSL session

     Done


sh dtlsProfile nsdtls_default_profile

    1)            Name: nsdtls_default_profile

                    PMTU Discovery: DISABLED

                    Max Record Size: 1459 bytes

                    Max Retry Time: 3 sec

                    Hello Verify Request: DISABLED

                    Terminate Session: DISABLED

                    Max Packet Count: 120 bytes


Done
<!--NeedCopy-->
```

DTLS cipher support

How to read the tables:

Unless a build number is specified, a cipher suite is supported for all builds in a release.

Example:

- **10.5, 11.0, 11.1, 12.0, 12.1**: All builds of 10.5, 11.0, 11.1, 12.0, 12.1 releases.

- **NA**: not applicable.

DTLS cipher support on NetScaler VPX, MPX/SDX (N2 and N3 based) appliances

Cipher Suite Name	Hex Code	Wireshark Cipher Suite Name	Builds Supported (frontend)	Builds Supported (backend)
TLS1-AES-256-CBC-SHA	0x0035	TLS_RSA_WITH_A	11.0, 11.1, 12.0, 12.1	12.0, 12.1
TLS1-AES-128-CBC-SHA	0x002f	TLS_RSA_WITH_AES_128_CBC_SHA	11.0, 11.1, 12.0, 12.1	12.0, 12.1
SSL3-DES-CBC-SHA	0x0009	TLS_RSA_WITH_D	11.0, 11.1, 12.0, 12.1	NA
SSL3-DES-CBC3-SHA	0x000a	TLS_RSA_WITH_3DES_EDE_CBC_SHA	11.0, 11.1, 12.0, 12.1	12.0, 12.1
SSL3-EDH-RSA-DES-CBC3-SHA	0x0016	TLS_DHE_RSA_WI	11.0, 11.1, 12.0, 12.1	NA
SSL3-EDH-RSA-DES-CBC-SHA	0x0015	TLS_DHE_RSA_WITH_DES_CBC_SHA	11.0, 11.1, 12.0, 12.1	NA
TLS1-ECDHE-RSA-AES256-SHA	0xc014	TLS_ECDHE_RSA_	12.1	12.1
TLS1-ECDHE-RSA-AES128-SHA	0xc013	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	12.1	12.1
TLS1-ECDHE-RSA-DES-CBC3-SHA	0xc012	TLS_ECDHE_RSA_	12.1	NA
TLS1-DHE-RSA-AES-128-CBC-SHA	0x0033	TLS_DHE_RSA_WITH_AES_128_CBC_SHA	12.1	12.1
TLS1-DHE-RSA-AES-256-CBC-SHA	0x0039	TLS_DHE_RSA_WI	12.1	12.1

Support for Intel Coleto SSL chip based platforms

September 21, 2020

The following platforms ship with Intel Coleto chips:

- MPX 5900
- MPX/SDX 8900
- MPX/SDX 26000-100G
- MPX/SDX 15000-50G

Use the 'show hardware' command to identify whether your appliance has Coleto (COL) chips.

```
sh hardware


    Platform: NSMPX-8900 8*CPU+4*F1X+6*E1K+1*E1K+1*COL 8955 30010


    Manufactured on: 10/18/2016


    CPU: 2100MHZ


    Host Id: 0


    Serial no: CRAC5CR8UA


    Encoded serial no: CRAC5CR8UA


 Done
<!--NeedCopy-->
```

Limitations:

- DTLS is not supported.
- DH 512 cipher is not supported.
- SSLv3 protocol is not supported.
- Hardware security module (HSM) is not supported.
- GnuTLS is not supported.

- ECDSA certificates with ECC curves P_224 and P521 are not supported (Not supported on platforms with Cavium chips also.)
- DNSSEC offload is not supported. (DNSSEC is supported in software but offload to hardware is not supported.)

MPX 9700/10500/12500/15500 FIPS appliances

September 21, 2020

The Federal Information Processing Standard (FIPS), issued by the US National Institute of Standards and Technologies, specifies the security requirements for a cryptographic module used in a security system. The NetScaler FIPS appliance complies with the second version of this standard, FIPS-140-2.

Note: Henceforth, all references to FIPS imply FIPS-140-2.

The FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—and a Cavium CN1620-NFBE3-2.0-G on the MPX 9700/10500/12500/15500 FIPS appliances—designed to comply with the FIPS 140-2 Level-2 specifications. The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (nsroot) can perform operations on the keys stored inside the HSM.

The following table summarizes the differences between standard NetScaler and NetScaler FIPS appliances.

Setting	NetScaler appliance	NetScaler FIPS appliance
Key storage	On the hard disk	On the FIPS card
Cipher support	All ciphers	FIPS approved ciphers
Accessing keys	From the hard disk	Not accessible

Configuring a FIPS appliance involves configuring the HSM immediately after completing the generic configuration process. You then create or import a FIPS key. After creating a FIPS key, you should export it for backup. You might also need to export a FIPS key so that you can import it to another appliance. For example, configuring FIPS appliances in a high availability (HA) setup requires transferring the FIPS key from the primary node to the secondary node immediately after completing the standard HA setup.

You can upgrade the firmware version on the FIPS card from version 4.6.0 to 4.6.1, and you can reset an

HSM that has been locked to prevent unauthorized logon. Only FIPS approved ciphers are supported on a NetScaler FIPS appliance.

HSM configuration

Before you can configure the HSM of your NetScaler FIPS appliance, you must complete the initial hardware configuration. For more information about MPX appliances, see Initial Configuration. For information about SDX appliances, click here.

Configuring the HSM of your NetScaler FIPS appliance erases all existing data on the HSM. To configure the HSM, you must be logged on to the appliance as the superuser (nsroot account). The HSM is preconfigured with default values for the Security Officer (SO) password and User password, which you use to configure the HSM or reset a locked HSM. The maximum length allowed for the password is 14 alphanumeric characters. Symbols are not allowed.

Important: Do not perform the
set ssl fips command without first resetting the FIPS card and restarting the MPX FIPS appliance.

Although the FIPS appliance can be used with the default password values, you should modify them before using it. The HSM can be configured only when you log on to the appliance as the superuser and specify the SO and User passwords.

Important: Due to security constraints, the appliance does not provide a means for retrieving the SO password. Store a copy of the password safely. Should you need to reinitialize the HSM, you will need to specify this password as the old SO password.

Before initializing the HSM, you can upgrade to the latest build of the software. To upgrade to the latest build, see Upgrading or Downgrading the System Software.

After upgrading, verify that the /nsconfig/fips directory has been successfully created on the appliance.

Configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the CLI

After logging on to the appliance as the superuser and completing the initial configuration, at the command prompt, type the following commands to configure the HSM and verify the configuration:

```
show ssl fips

reset ssl fips

reboot

set ssl fips -initHSM Level-2 <newSOpassword> <oldSOpassword> <
   userPassword> [-hsmLabel <string>]
```

```

save ns config

reboot

show ssl fips
<!--NeedCopy-->
```

Example:

```
show fips

    FIPS Card is not configured

    Done

    reset fips

    reboot

    Are you sure you want to restart NetScaler (Y/N)? [N]:y

    set ssl fips -initHSM Level-2 sopin12345 so12345 user123 -hsmLabel
        cavium

    This command will erase all data on the FIPS card. You must save
        the configuration

    (saveconfig) after executing this command.


    Do you want to continue?(Y/N)y

    Done

    save ns config

    reboot

    Are you sure you want to restart NetScaler (Y/N)? [N]:y

    show fips

            FIPS HSM Info:

```

```
HSM Label              : NetScaler FIPS

Initialization         : FIPS-140-2 Level-2

HSM Serial Number      : 2.1G1008-IC000021

HSM State              : 2

HSM Model              : NITROX XL CN1620-NFBE

Firmware Version       : 1.1

Firmware Release Date  : Jun04,2010


Max FIPS Key Memory    : 3996

Free FIPS Key Memory   : 3994

Total SRAM Memory      : 467348

Free SRAM Memory       : 62564

Total Crypto Cores      : 3

Enabled Crypto Cores    : 1

Done

Note: If you upgrade the firmware to version 2.2, the firmware release date is replaced with the firmware build.



show fips

FIPS HSM Info:

HSM Label                : NetScaler FIPS

Initialization                : FIPS-140-2 Level-2

HSM Serial Number    : 3.0G1235-ICM000264

HSM State                : 2
```

```

    HSM Model               : NITROX XL CN1620-NFBE

    Hardware Version       : 2.0-G

    Firmware Version        : 2.2

    Firmware Build           : NFBE-FW-2.2-130009



    Max FIPS Key Memory : 3996

    Free FIPS Key Memory : 3958

    Total SRAM Memory    : 467348

    Free SRAM Memory     : 50524

    Total Crypto Cores      : 3

    Enabled Crypto Cores  : 3

    Done
<!--NeedCopy-->
```

Configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the GUI

1. Navigate to Traffic Management > SSL > FIPS.
2. In the details pane, on the FIPS Infotab, click Reset FIPS.
3. In the navigation pane, click System.
4. In the details pane, click Reboot.
5. In the details pane, on the FIPS Info tab, click Initialize HSM.
6. In the Initialize HSM dialog box, specify values for the following parameters:
    - Security Officer (SO) Password*—new SO password
    - Old SO Password*—old SO password
    - User Password*—user password
    - Level—initHSM (Currently set to Level2 and cannot be changed)
    - HSM Label—hsmLabel

    *A required parameter

7. Click OK.
8. In the details pane, click Save.
9. In the navigation pane, click System.
10. In the details pane, click Reboot.
11. Under FIPS HSM Info, verify that the information displayed for the FIPS HSM that you just configured is correct.

Create and transfer FIPS keys

After configuring the HSM of your FIPS appliance, you are ready to create a FIPS key. The FIPS key is created in the appliance's HSM. You can then export the FIPS key to the appliance's CompactFlash card as a secured backup. Exporting the key also enables you to transfer it by copying it to the /flash of another appliance and then importing it into the HSM of that appliance. You must enable SIM between two standalone nodes before you export and transfer the keys. In an HA setup, if one of the nodes is replaced with a new appliance, you must enable SIM between this new appliance and the existing appliance of the HA setup before you export or import FIPS keys.

Instead of creating a FIPS key, you can import an existing FIPS key or import an external key as a FIPS key. If you are adding a certificate-key pair of 2048 bits on the MPX 9700/10500/12500/15500 FIPS appliances, make sure that you have the correct certificate and key pair.

Note: If you are planning an HA setup, make sure that the FIPS appliances are configured in an HA setup before creating a FIPS key.

Create FIPS keys

Before creating a FIPS key, make sure that the HSM is configured.

You must specify the key type (RSA or ECDSA) and specify the curve for ECDSA keys.

Create a FIPS key by using the GUI

1. Navigate to Traffic Management > SSL > FIPS.
2. In the details pane, on the FIPS Keys tab, click Add.
3. In the Create FIPS Key dialog box, specify values for the following parameters:
    - FIPS Key Name*—fipsKeyName
    - Modulus*—modulus
    - Exponent*—exponent

   *A required parameter

4. Click Create, and then click Close.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just created are correct.

Create a FIPS key by using the CLI

At the command prompt, type the following commands to create a FIPS key and verify the settings:

```
create ssl fipsKey <fipsKeyName> -modulus <positive_integer> [-exponent
     ( 3 | F4 )]

show ssl fipsKey [<fipsKeyName>]
<!--NeedCopy-->
```

Example:

```
    create fipskey Key-FIPS-1 -keytype RSA -modulus 2048 -exponent 3

    show ssl fipsKey Key-FIPS-1

                FIPS Key Name: Key-FIPS-1 Key Type: RSA Modulus: 2048
        Public Exponent: F4 (Hex: 0x10001)
<!--NeedCopy-->
```

Export FIPS keys

Citrix recommends that you create a backup of any key created in the FIPS HSM. If a key in the HSM is deleted, there is no way to create the same key again, and all the certificates associated with it are rendered useless.

In addition to exporting a key as a backup, you might need to export a key for transfer to another appliance.

The following procedure provides instructions on exporting a FIPS key to the /nsconfig/ssl folder on the appliance's CompactFlash and securing the exported key by using a strong asymmetric key encryption method.

Export a FIPS key by using the CLI

At the command prompt, type:

`export ssl fipsKey <fipsKeyName> -key <string>`

Example:

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
```

Export a FIPS key by using the GUI

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Keys tab, click Export.
3. In the Export FIPS key to a file dialog box, specify values for the following parameters:
    - FIPS Key Name*—fipsKeyName
    - File Name*—key (To put the file in a location other than the default, you can either specify the complete path or click the Browse button and navigate to a location.)

    *A required parameter
4. Click Export, and then click Close.

Import an existing FIPS key

To use an existing FIPS key with your FIPS appliance, you need to transfer the FIPS key from the hard disk of the appliance into its HSM.

Note: To avoid errors when importing a FIPS key, make sure that the name of the key imported is the same as the original key name when it was created.

Import a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the CLI

At the command prompt, type the following commands to import a FIPS key and verify the settings:

```
-  import ssl fipsKey <fipsKeyName> -key <string> -inform SIM -exponent
    (F4 | 3)
-  show ssl fipskey <fipsKeyName>
<!--NeedCopy-->
```

Example:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
show ssl fipskey key-FIPS-2
FIPS Key Name: Key-FIPS-2 Modulus: 2048   Public Exponent: F4 (Hex
    value 0x10001)
<!--NeedCopy-->
```

Import a FIPS key by using the GUI

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Keys tab, click Import.
3. In the Import as a FIPS Key dialog box, select FIPS key file and set values for the following parameters:
    - FIPS Key Name*
    - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.
    - Exponent*

    *A required parameter
4. Click Import, and then click Close.
5. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just imported are correct.

Import an external key

In addition to transferring FIPS keys that are created within the NetScaler appliance's HSM, you can transfer external private keys (such as those created on a standard NetScaler, Apache, or IIS) to a FIPS NetScaler appliance. External keys are created outside the HSM, by using a tool such as OpenSSL. Before importing an external key into the HSM, copy it to the appliance's flash drive under /nsconfig/ssl.

On the MPX 9700/10500/12500/15500 FIPS appliances, the -exponent parameter in the import ssl fipskey command is not required while importing an external key. The correct public exponent is detected automatically when the key is imported, and the value of the -exponent parameter is ignored.

The NetScaler FIPS appliance does not support external keys with a public exponent other than 3 or F4.

You do not need a wrap key on the MPX 9700/10500/12500/15500 FIPS appliances.

You cannot import an external, encrypted FIPS key directly to an MPX 9700/10500/12500/15500 FIPS appliance. To import the key you need to first decrypt the key, and then import it. To decrypt the key, at the shell prompt, type:

```
openssl rsa -in <EncryptedKey.key> > <DecryptedKey.out>
<!--NeedCopy-->
```

Import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the CLI

1. Copy the external key to the appliance's flash drive.
2. If the key is in .pfx format, you must first convert it to PEM format. At the command prompt, type:

   ```
   convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx file name> -password <password>
   <!--NeedCopy-->
   ```

3. At the command prompt, type the following commands to import the external key as a FIPS key and verify the settings:

   ```
   import ssl fipsKey <fipsKeyName> -key <string> -informPEM`
   show ssl fipskey<fipsKeyName>
   <!--NeedCopy-->
   ```

Example:

```
convert ssl pkcs12 iis.pem -password 123456 -import -pkcs12File iis.pfx

import fipskey Key-FIPS-2 -key iis.pem -inform PEM

show ssl fipskey key-FIPS-2

FIPS Key Name: Key-FIPS-2 Modulus: 0   Public Exponent: F4 (Hex value 0x10001)
<!--NeedCopy-->
```

Import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the GUI

1. If the key is in .pfx format, you must first convert it to PEM format.
   a) Navigate to **Traffic Management** > **SSL**.
   b) In the details pane, under Tools, click Import PKCS#12.
   c) In the Import PKCS12 File dialog box, set the following parameters:
      - Output File Name*
      - PKCS12 File Name*—Specify the .pfx file name.
      - Import Password*
      - Encoding Format

      *A required parameter
2. Navigate to **Traffic Management** > **SSL** > **FIPS**.
3. In the details pane, on the FIPS Keys tab, click Import.

4. In the Import as a FIPS Key dialog box, select PEM file, and set values for the following parameters:

    - FIPS Key Name*
    - Key File Name*—To put the file in a location other than the default, you can either specify the complete path or click Browse and navigate to a location.

    *A required parameter

5. Click Import, and then click Close.
6. On the FIPS Keys tab, verify that the settings displayed for the FIPS key that you just imported are correct.

Configure FIPS appliances in a high availability setup

September 21, 2020

You can configure two appliances in a high availability (HA) pair as FIPS appliances. For information about configuring an HA setup, see High availability.

Note: Citrix recommends that you use the configuration utility (GUI) for this procedure. If you use the command line (CLI), make sure that you carefully follow the steps as listed in the procedure. Changing the order of steps or specifying an incorrect input file might cause an inconsistency that requires you to restart the appliance. In addition, if you use the CLI, the `create ssl fipskey` command is not propagated to the secondary node. When you run the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not identical. Create the FIPS key on one of the nodes and then transfer it to the other node. But if you use the configuration utility to configure FIPS appliances in an HA setup, the FIPS key that you create is automatically transferred to the secondary node. The process of managing and transferring the FIPS keys is known as secure information management (SIM).

> **Important:** On the MPX 9700/10500/12500/15500 FIPS appliances, the HA setup must be completed within six minutes. If the process takes longer than six minutes, the internal timer of the FIPS card expires and the following error message appears:
>
> ```
> ERROR: Operation timed out or repeated, please wait for 10 mins and redo
>  the SIM/HA configuration steps.
> ```

If this message appears, restart the appliance or wait for 10 minutes, and then repeat the HA setup procedure.

In the following procedure, appliance A is the primary node and appliance B is the secondary node.

Configure FIPS appliances in a high availability setup by using the CLI

1. **On appliance A**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on to the appliance, using the administrator credentials.
3. Initialize appliance A as the source appliance. At the command prompt, type:

   ```
   init ssl fipsSIMsource <certFile>
   <!--NeedCopy-->
   ```

   Example:

   ```
   init fipsSIMsource /nsconfig/ssl/nodeA.cert
   ```

4. Copy this `<certFile>` file to appliance B, in the /nconfig/ssl folder.

   Example:

   ```
   scp /nsconfig/ssl/nodeA.cert nsroot@198.51.100.10:/nsconfig/ssl
   ```

5. **On appliance B**, open an SSH connection to the appliance by using an SSH client, such as PuTTY.
6. Log on to the appliance, using the administrator credentials.
7. Initialize appliance B as the target appliance. At the command prompt, type:

   ```
   init ssl fipsSIMtarget <certFile> <keyVector> <targetSecret>
   <!--NeedCopy-->
   ```

   Example:

   ```
   init fipsSIMtarget /nsconfig/ssl/nodeA.cert /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeB.secret
   ```

8. Copy this `<targetSecret>` file to appliance A.

   Example:

   ```
   scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.20:/nsconfig/ssl
   ```

9. **On appliance A**, enable appliance A as the source appliance. At the command prompt, type:

   ```
   enable ssl fipsSIMSource <targetSecret> <sourceSecret>
   <!--NeedCopy-->
   ```

   Example:

   ```
   enable fipsSIMsource /nsconfig/ssl/nodeB.secret /nsconfig/ssl/nodeA.secret
   ```

10. Copy this <`sourceSecret`> file to appliance B.

    Example:

    ```
    scp /nsconfig/ssl/fipslbdal0801b.secret nsroot@198.51.100.10:/nsconfig/ssl
    ```

11. **On appliance B**, enable appliance B as the target appliance. At the command prompt, type:

    ```
    enable ssl fipsSIMtarget <keyVector> <sourceSecret>
    <!--NeedCopy-->
    ```

    Example:

    ```
    enable fipsSIMtarget /nsconfig/ssl/nodeB.key /nsconfig/ssl/nodeA.secret
    ```

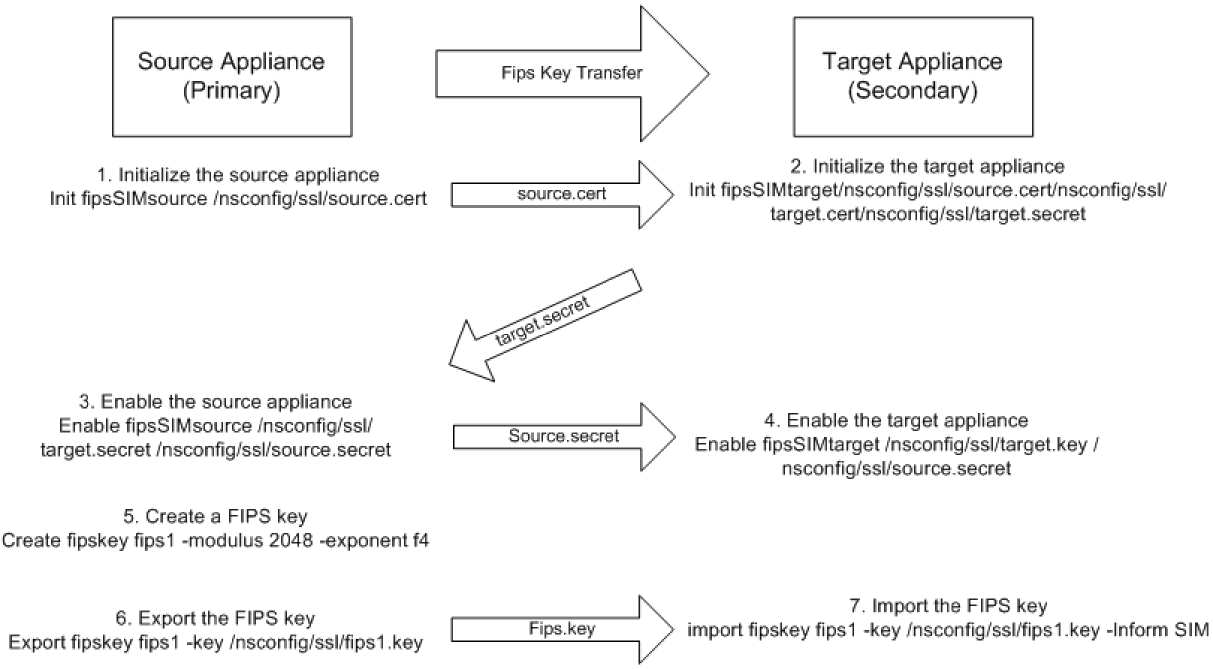
12. **On appliance A**, create a FIPS key, as described in Create a FIPS key.
13. Export the FIPS key to the appliance's hard disk, as described in Export a FIPS key.
14. Copy the FIPS key to the hard disk of the secondary appliance by using a secure file transfer utility, such as SCP.
15. **On appliance B**, import the FIPS key from the hard disk into the HSM of the appliance, as described in Import an existing FIPS key.

Configure FIPS appliances in a high availability setup by using the GUI

1. On the appliance to be configured as the source appliance, navigate to **Traffic Management > SSL > FIPS**.
2. In the details pane, on the FIPS Info tab, click **Enable SIM**.
3. In the **Enable HA Pair for SIM** dialog box, go to the **Certificate File Name** text box. Type the file name, with the path to the location at which the FIPS certificate must be stored on the source appliance.
4. In the **Key Vector File Name** text box, type the file name, with the path to the location at which the FIPS key vector must be stored on the source appliance.
5. In the **Target Secret File Name** text box, type the location for storing the secret data on the target appliance.
6. In the **Source Secret File Name** text box, type the location for storing the secret data on the source appliance.
7. Click **OK**. The FIPS appliances are now configured in HA mode.
8. Create a FIPS key, as described in Create a FIPS key. The FIPS key is automatically transferred from the primary to the secondary.

The following diagram summarizes the transfer process.

Figure 1. Transfer the FIPS key-summary



1. Initialize the source appliance
Init fipsSIMsource /nsconfig/ssl/source.cert

2. Initialize the target appliance
Init fipsSIMtarget/nsconfig/ssl/source.cert/nsconfig/ssl/target.cert/nsconfig/ssl/target.secret

3. Enable the source appliance
Enable fipsSIMsource /nsconfig/ssl/target.secret /nsconfig/ssl/source.secret

4. Enable the target appliance
Enable fipsSIMtarget /nsconfig/ssl/target.key /nsconfig/ssl/source.secret

5. Create a FIPS key
Create fipskey fips1 -modulus 2048 -exponent f4

6. Export the FIPS key
Export fipskey fips1 -key /nsconfig/ssl/fips1.key

7. Import the FIPS key
import fipskey fips1 -key /nsconfig/ssl/fips1.key -Inform SIM

Update the firmware to version 2.2 on a FIPS card

September 21, 2020

FIPS firmware version 2.2 supports TLS protocol versions 1.1 and 1.2. From the command line, you can update the firmware version of the FIPS card of a NetScaler MPX 9700/10500/12500/15500 FIPS appliance from version 1.1 to version 2.2.

For successful SIM key propagation from primary to secondary in a high availability (HA) pair, the Cavium firmware version on each appliance should be identical. Perform the firmware update on the secondary appliance first. If executed on the primary appliance first, the long-running update process causes a failover.

Limitations

- Secure renegotiation is supported only on SSL virtual servers and front-end SSL services.
- Creating a certificate signing request by using a key that was created on firmware version 1.1 and updated to firmware version 2.2 fails.
- You cannot create a 1024-bit RSA key on firmware version 2.2. However, if you have imported or created a 1024-bit FIPS key on firmware version 1.1 and you then update to firmware version 2.2, you can use that FIPS key on firmware version 2.2.
- 1024-bit RSA keys are not supported.
- Secure renegotiation using SSLv3 protocol is not supported.

- After you upgrade the firmware, TLSv1.1 and TLSv1.2 are disabled by default on the existing virtual server, internal, front end, and backend services. To use TLS 1.1/1.2, you must explicitly enable these protocols, on the SSL entities, after the upgrade.
- FIPS keys that are created in firmware version 2.2 are not available if you downgrade the firmware to version 1.1.

Prerequisites

Download the following files from the download page on www.citrix.com. The files must be stored in the /var/nsinstall directory on the appliance.

- FW 2.2 File: FW-2.2-130013
- FW 2.2 Signature File: FW-2.2-130013.sign

FW-2.2-130013 is the recommended firmware version. It includes fixes to improve DRBG.

Update the FIPS firmware to version 2.2 on a standalone appliance

1. Log on to the appliance by using the administrator credentials.
2. At the prompt, type the following command to confirm that the FIPS card is initialized.

```
show fips

FIPS HSM Info:
HSM Label       : NetScaler FIPS
Initialization      : FIPS-140-2 Level-2
HSM Serial Number   : 3.0G1235-ICM000264
HSM State       : 2
HSM Model       : NITROX XL CN1620-NFBE

Hardware Version    : 2.0-G
Firmware Version    : 1.1
Firmware Release Date   : Jun04,2010

Max FIPS Key Memory : 3996
Free FIPS Key Memory    : 3992
Total SRAM Memory   : 467348
Free SRAM Memory    : 62512
Total Crypto Cores  : 3
Enabled Crypto Cores    : 1
Done
<!--NeedCopy-->
```

3. Save the configuration. At the prompt, type:

```
save config
<!--NeedCopy-->
```

4. Perform the update. At the prompt, type:

   ```
   update ssl fips -fipsFW <path to the extracted contents>/CN16XX-NFBE-FW-2.2-1300013
   <!--NeedCopy-->
   ```

   Press Y when the following prompt appears:

   ```
   This command will update compatible version of the FIPS firmware. You must save the current configuration (saveconfig) before executing this command. You must reboot the system after execution of this command, for the firmware update to take effect. Do you want to continue?(Y/N)Y

   Done
   <!--NeedCopy-->
   ```

Note: You only need to specify the firmware file, because the firmware signature file is placed in the same location.

The update takes up to ten seconds. The update command is blocking, which means that no other actions are executed until the command finishes. The command prompt reappears when execution of the command is completed.

1. Restart the appliance. At the prompt, type:

   ```
   reboot

   Are you sure you want to restart NetScaler (Y/N)? [N]:Y
   <!--NeedCopy-->
   ```

2. Verify that the update is successful. At the prompt, type:

   ```
   show fips
   <!--NeedCopy-->
   ```

   The firmware version displayed in the output should be 2.2. For example:

   ```
   sh fips
       FIPS HSM Info:
           HSM Label        : NetScaler FIPS
           Initialization       : FIPS-140-2 Level-2
   ```

```
        HSM Serial Number   : 2.1G1207-IC002429
        HSM State       : 2
        HSM Model       : NITROX XL CN1620-NFBE

        Hardware Version    : 2.0-G
        Firmware Version    : 2.2
        Firmware Build         : NFBE-FW-2.2-130013
        Max FIPS Key Memory : 3996
        Free FIPS Key Memory    : 3982
        Total SRAM Memory   : 467348
        Free SRAM Memory    : 50472
        Total Crypto Cores  : 3
        Enabled Crypto Cores    : 1
 Done
<!--NeedCopy-->
```

Update the FIPS firmware to version 2.2 on appliances in a high availability pair

1. Log on to the secondary node and perform the update as described in [“Update the FIPS firmware to version 2.2 on a standalone appliance”.

   Force the secondary node to become primary. At the prompt, type:

   ```
   force failover
   <!--NeedCopy-->
   ```

   Press **Y** at the confirmation prompt.

2. Log on to the new secondary node (old primary) and perform the update as described in “Update the FIPS firmware to version 2.2 on a standalone appliance”.

3. Force the new secondary node to become primary again. At the prompt, type:

   ```
   force failover
   <!--NeedCopy-->
   ```

   Press **Y** at the confirmation prompt.

Update the FIPS firmware to version 1.1 on a standalone appliance

1. Download the nfb_firmware-r1235_100604 and nfb_firmware-r1235_100604.sign files, to the same directory on the appliance, from the download page on www.citrix.com.

2. Log on to the appliance by using the administrator credentials.

3. At the prompt, type:

```
update ssl fips -fipsFW /<full path to the file>/nfb_firmware-
    r1235_100604
<!--NeedCopy-->
```

Reset a locked HSM

September 21, 2020

The HSM becomes locked (no longer operational) if you change the SO password, restart the appliance without saving the configuration, and make three unsuccessful attempts to change the password. This is a security measure for preventing unauthorized access attempts and changes to the HSM settings.

Important: To avoid this situation, save the configuration after initializing the HSM.
If the HSM is locked, you must reset the HSM and restart the appliance to restore the default passwords. You can then use the default passwords to access the HSM and configure it with new passwords. When finished, you must save the configuration and restart the appliance.

Caution: Do not reset the HSM unless it has become locked.

Reset a locked HSM by using the CLI

At the command prompt, type the following commands to reset and re-initialize a locked HSM:

```
reset ssl fips
reboot -warm
set ssl fips -initHSM Level-2 <new SO password> <old SO password> <user
    password> [-hsmLabel <string>]
save ns config
reboot -warm
<!--NeedCopy-->
```

Example:

```
reset fips

reboot -warm

set fips -initHSM Level-2 newsopin123 sopin123 userpin123 -hsmLabel
   NSFIPS

saveconfig
```

```

reboot -warm

Note: By default the HSM passwords are preconfigured. The <
    Old_SO_Password> = so12345, <User_Password> = user123, <
    New_SO_Password> = sopin12345, <New_User_Password> = userpin123.
<!--NeedCopy-->
```

Reset a locked HSM by using the GUI

1. Navigate to Traffic Management > SSL > FIPS
2. In the details pane, on the FIPS Info tab, click Reset FIPS.
3. Configure the HSM, as described in Configuring the HSM.
4. In the details pane, click Save.

MPX 14000 FIPS appliances

September 21, 2020

> **Important:**
>
> Configuration steps for NetScaler MPX 14000 FIPS and NetScaler MPX 9700/10500/12500/15500 FIPS appliances are different. MPX 14000 FIPS appliances do not use firmware v2.2. A FIPS key created on the HSM of MPX 9700 platform cannot be transferred to HSM of MPX 14000 platform and vice a versa. However, if you have imported an RSA key as a FIPS key, you can copy the RSA key to the MPX 14000 platform, and then import it as a FIPS key. Only 2048-bit and 3072-bit keys are supported.

A FIPS appliance is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CNN3560-NFBE-G—designed to comply with the FIPS 140-2 Level-3 specifications (from release 12.0 build 56.x). The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser (`nsroot`) can perform operations on the keys stored inside the HSM.

Before configuring a FIPS appliance, you must check the state of the FIPS card and then initialize the card. Create a FIPS key and server certificate, and add any additional SSL configuration.

For information about the FIPS ciphers supported, see FIPS Approved Algorithms and Ciphers.

For information about configuring FIPS appliances in a high availability setup, see Configuring FIPS Appliances in a High Availability Setup.

Limitations

1. SSL renegotiation using the SSLv3 protocol is not supported on the back end of an MPX FIPS appliance.
2. 1024-bit and 4096-bit keys and exponent value of 3 are not supported.

Configure the HSM

Before configuring the HSM on an MPX 14000 FIPS appliance, you must check the state of your FIPS card to verify that the driver loaded correctly, and then initialize the card.

At the command prompt, type:

```
show fips

FIPS Card is not configured

Done
<!--NeedCopy-->
```

The message "ERROR: Operation not permitted - no FIPS card present in the system" appears if the driver is not loaded correctly.

Initialize the FIPS card

The appliance must be restarted three times for proper initialization of the FIPS card.

> **Important**
>
> - Verify that the /nsconfig/fips directory has been successfully created on the appliance.
> - Do not save the configuration before you restart the appliance for the third time.

Perform the following steps to initialize the FIPS card:

1. Reset the FIPS card.
2. Restart the appliance.
3. Set the security officer password for partitions 0 and 1, and the user password for partition 1. Note: The set or reset command takes more than 60 seconds to run.
4. Save the configuration.
5. Verify that the password encrypted key for the master partition (master_pek.key) has been created in the /nsconfig/fips/ directory.
6. Restart the appliance.
7. Verify that the password encrypted key for the default partition (default_pek.key) has been created in the /nsconfig/fips/ directory.

8. Restart the appliance.
9. Verify that the FIPS card is UP.

Initialize the FIPS card by using the CLI

At the command prompt, type the following commands:

```
reset fips
Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. Do you want
     to continue?(Y/N)y

Done
<!--NeedCopy-->
```

Note: The following message appears when you run the **set fips** command:

```
This command will erase all data on the FIPS card. You must save the
    configuration (saveconfig) after executing this command. [Note: On
    MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by
    default, and the -initHSM Level-2 option is internally converted to
    Level-3]  Do you want to continue?(Y/N)y

saveconfig
Done

reboot

reboot

show fips

        FIPS HSM Info:

                 HSM Label              : NSFIPS

                 Initialization         : FIPS-140-2 Level-2

                 HSM Serial Number      : 3.0G1501-ICM000083

                 HSM State              : 2

                 HSM Model              : No Model Info

                 Hardware Version       : 0.0-G

                 Firmware Version       : 1.0

                 Firmware Build         : NFBE-FW-1.0-39

                 Max FIPS Key Memory    : 102235

                 Free FIPS Key Memory   : 102233

                 Total SRAM Memory      : 557396

                 Free SRAM Memory       : 255456

                 Total Crypto Cores     : 63

                 Enabled Crypto Cores   : 63

Done
<!--NeedCopy-->
```

Create FIPS keys

You can create a FIPS key on your MPX 14000 FIPS appliance or import an existing FIPS key to the appliance. The MPX 14000 FIPS appliance supports only 2048-bit and 3072-bit keys and an exponent value of F4. For PEM keys, an exponent is not required. Verify that the FIPS key is created correctly. Create a certificate signing request and a server certificate. Finally, add the certificate-key pair to your appliance.

You must specify the key type (RSA or ECDSA) and specify the curve for ECDSA keys.

> **Note**
>
> 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.

Create a FIPS key by using the CLI

At the command prompt, type:

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (
    3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

Done

show ssl fipskey ddvws

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
    Hex: 0x10001)

Done
<!--NeedCopy-->
```

Import a FIPS key by using the CLI

At the command prompt, type:

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
    wrapKeyName <string>] [-iv<string>] -exponent  F4 ]
<!--NeedCopy-->
```

Example:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
Done
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
Done
<!--NeedCopy-->
```

Verify that the FIPS key is created or imported correctly by running the `show fipskey` command.

```
show fipskey
1)       FIPS Key Name: Key-FIPS-2
Done
<!--NeedCopy-->
```

Create a certificate signing request by using the CLI

At the command prompt, type:

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string> [-organizationUnitName <string>] [-localityName <string>] [-commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

Example:

```
create certreq f1.req  - fipsKeyName  f1 -countryName US  -stateName CA -organizationName Citrix -companyName Citrix -commonName ctx -emailAddress test@example.com
Done
<!--NeedCopy-->
```

Create a server certificate by using the CLI

At the command prompt, type:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <input_filename>][-CAkeyForm ( DER | PEM )] [-CAserial <output_filename>]
<!--NeedCopy-->
```

Example:

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-root.key -CAserial ns-root.srl -days 1000
Done
<!--NeedCopy-->
```

The preceding example creates a server certificate using a local root CA on the appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <
    string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>][-
    expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

Example:

```
add certkey cert1 -cert f1.cert -fipsKey f1
Done
<!--NeedCopy-->
```

After creating the FIPS key and server certificate, you can add the generic SSL configuration. Enable the features that are required for your deployment. Add servers, services, and SSL virtual servers. Bind the certificate-key pair and the service to the SSL virtual server. Save the configuration.

```
enable ns feature SSL LB
Done
add server s1 10.217.2.5
Done
add service sr1 s1 HTTP 80
Done
add lb vserver v1 SSL 10.217.2.172 443
Done
bind ssl vserver v1  - certkeyName cert1
Done
bind lb vserver v1 sr1
Done
saveconfig
Done
<!--NeedCopy-->
```

The basic configuration of your MPX 14000 FIPS appliance is now complete.

For information about configuring secure HTTPS, click here.

For information about configuring secure RPC, click here.

SDX 14000 FIPS appliances

September 21, 2020

A NetScaler SDX appliance is a multitenant platform on which you can provision and manage multiple virtual NetScaler instances. The SDX appliance addresses cloud computing and multitenancy requirements by allowing a single administrator to configure and manage the appliance and delegate the administration of each hosted instance to tenants.

A NetScaler SDX 14030/14060/14080 FIPS appliance provides the capabilities of an SDX appliance with FIPS functionality. It is equipped with a tamper-proof (tamper-evident) cryptographic module—a Cavium CNN3560-NFBE-G—designed to comply with the FIPS 140-2 Level-3 specifications (from release 12.0 build 56.x). The Critical Security Parameters (CSPs), primarily the server's private-key, are securely stored and generated inside the cryptographic module, also referred to as the Hardware Security Module (HSM). The CSPs are never accessed outside the boundaries of the HSM. Only the superuser can perform operations on the keys stored inside the HSM.

A NetScaler SDX 14030/14060/14080 FIPS appliance contains one FIPS HSM module with 63 cores. The FIPS HSM module can be partitioned up to a maximum of 32 partitions. The SDX administrator can assign dedicated key storage, cryptographic resources, and number of crypto SSL FIPS cores to each partition. Keys and resources allocated to a partition are dedicated and secure and cannot be accessed or shared by another partition.

The FIPS HSM partition that you create can be assigned or attached to a VPX instance at the time of provisioning the instance, or later by editing the instance. The FIPS partition created and attached to an instance acts like a virtual HSM module for that instance.

The VPX instances on an SDX 14030/14060/14080 FIPS appliance are assigned a FIPS virtual function (VF) partition, which is treated as an isolated FIPS virtual card or HSM. Therefore, the steps to configure a FIPS partition inside a VPX instance are similar to the steps to configure an MPX FIPS appliance. For compliance details, see the security policy details on the U.S. National Institute of Standards and Technology (NIST) website.

For information about configuring FIPS appliances in a high availability setup, see FIPS Appliances in a High Availability Setup.

> **Important**
>
> Each key includes a private and a public key. As a result, it occupies two key spaces. Therefore, the maximum number of keys is limited to one less than half the key store size.

The SDX 14000 FIPS platform supports a hybrid FIPS mode. In this mode, you can offload part of the encryption and decryption activity to a non-FIPS card. For more information, see Hybrid FIPS mode.

Limitations

September 21, 2020

1. SSL renegotiation using the SSLv3 protocol is not supported on the back end of an SDX FIPS appliance.
2. 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.
3. Backup and restore is not supported.
4. Cluster and administrative domains are not supported.
5. You can attach only one FIPS partition to an instance.
6. An instance with a FIPS partition can be assigned only one CPU core.
7. You can assign either a FIPS partition or an SSL core to an instance, but not both.

Terminology

September 21, 2020

Zeroize: Reset the HSM. All the data on the HSM is deleted. This is a mandatory step before the HSM is initialized.

Initialize: Set the HSM capabilities. The NetScaler SDX FIPS appliance complies with FIPS-140-2 level 2. You can create partitions after you initialize the chip.

Key store size: Number of keys that can be stored on a partition. A maximum of 102235 keys can be specified. The maximum number of keys that can be stored is one less than half the number specified. For example, if you specify 100, you can create only 49 keys because one of the keys is the RSA key pair that consumes 2 key stores.

Crypto Core Capacity: Number of crypto cores assigned to a partition. A maximum of 63 cores are available.

SSL Context: Number of concurrent SSL connections that can be created on a partition.

Initialize the HSM

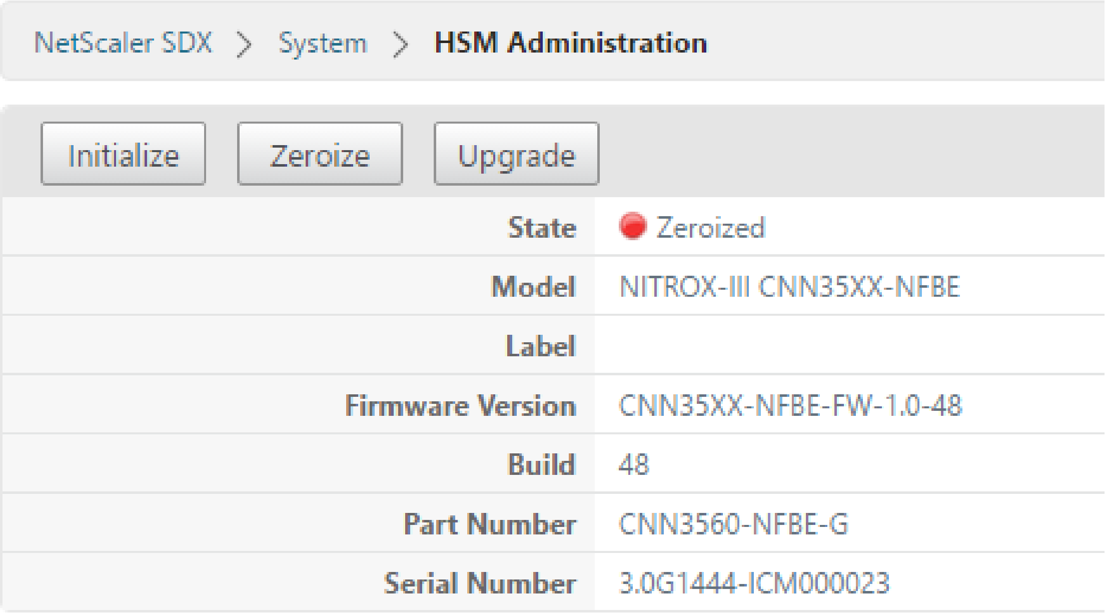
September 21, 2020

Before initializing the HSM, you must first zeroize it.

Zeroize the HSM by using the Management Service

1. Open a browser and log on to the appliance.
2. On the **Configuration** tab, navigate to **System** > **HSM Administration**, and in the details plane, click **Zeroize**.

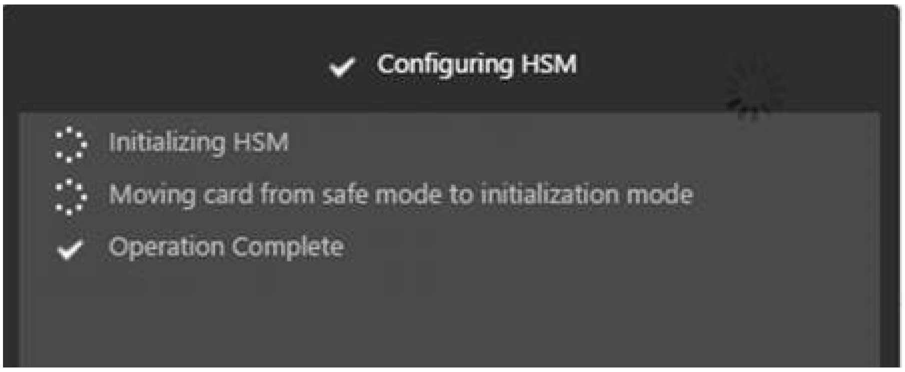
   All data is wiped from the FIPS chip, and the state appears as "Zeroized." Any HSM partitions created earlier are deleted.



State	● Zeroized
Model	NITROX-III CNN35XX-NFBE
Label	
Firmware Version	CNN35XX-NFBE-FW-1.0-48
Build	48
Part Number	CNN3560-NFBE-G
Serial Number	3.0G1444-ICM000023

Initialize the HSM by using the Management Service

1. On the **Configuration** tab, navigate to **System** > **HSM Administration**, and in the details plane, click **Initialize**.
2. Type a new user name, specify a password, and click **OK**.



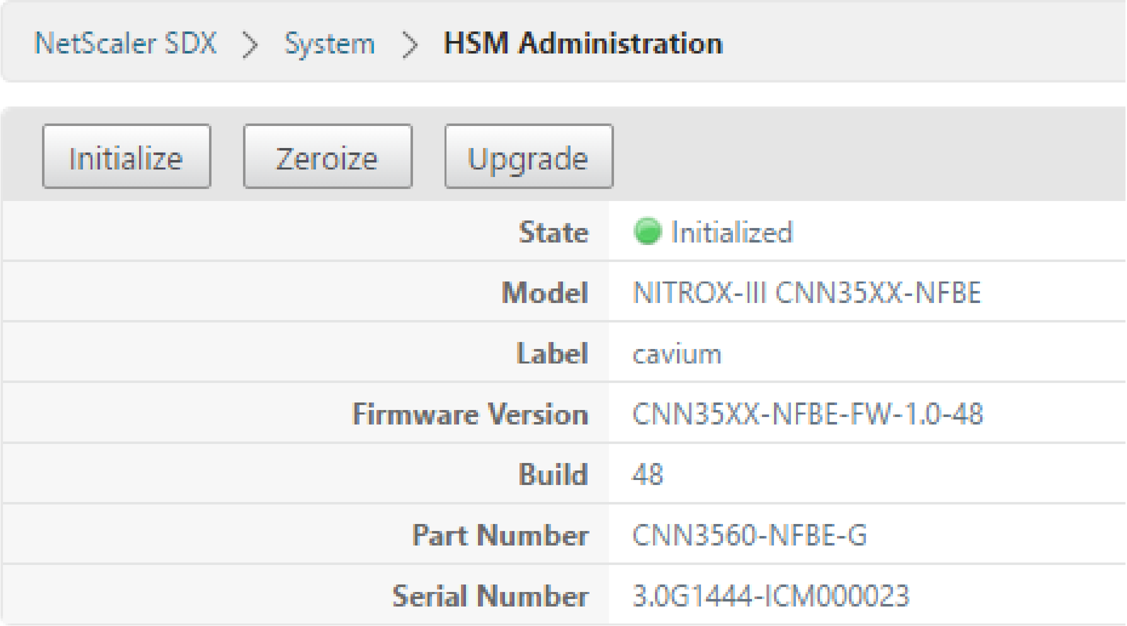
✓ Configuring HSM

Initializing HSM

Moving card from safe mode to initialization mode

✓ Operation Complete

The card state appears as "Initialized."



State	● Initialized
Model	NITROX-III CNN35XX-NFBE
Label	cavium
Firmware Version	CNN35XX-NFBE-FW-1.0-48
Build	48
Part Number	CNN3560-NFBE-G
Serial Number	3.0G1444-ICM000023

Create partitions

September 21, 2020

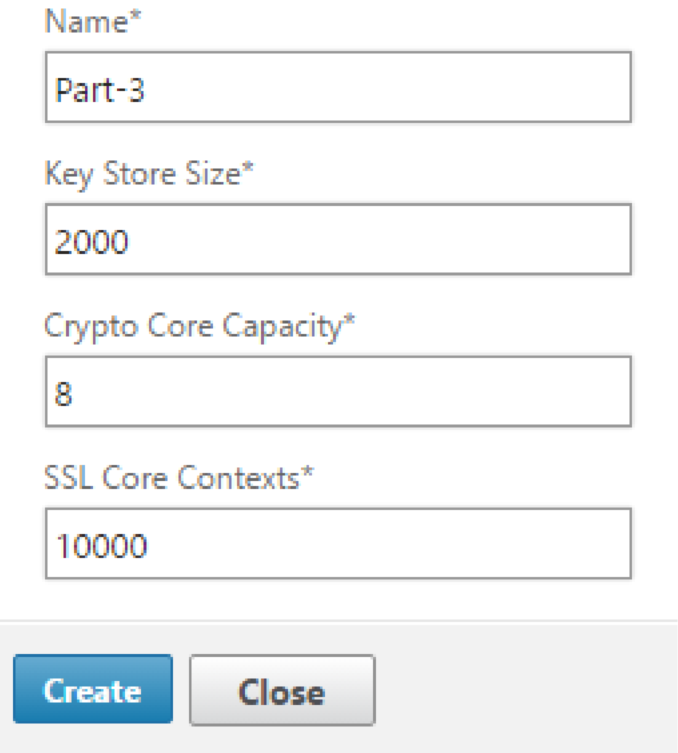
Create partitions for different tenants and specify the cryptographic resources for each partition. Each instance is assigned one partition, and a partition can be assigned to only one instance. Deleting an instance deletes the partition assigned to the instance. As a result, the partition data is also deleted and not left unsecured or accessible later. Number of keys and SSL context assignment depends on your application. For information about number of cores to assign, see the NetScaler datasheet.

> **Important**
>
> After you assign a key store size and cores to an HSM partition, you cannot change them at run time. You must first detach the partition from the instance.

Create a partition by using the Management Service

1. On the Configuration tab, navigate to **System** > **HSM Administration** > **Partitions**, and in the details plane, click **Add**.
2. Specify a name for the partition, and the resources to be assigned to this partition.

3. Click **OK**.



Name*: Part-3

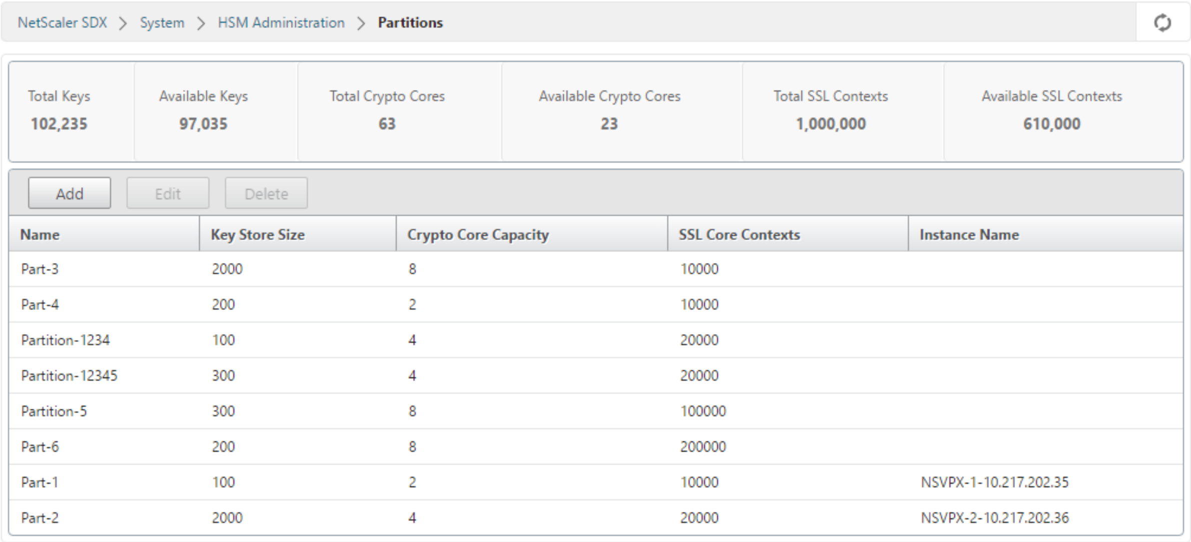
Key Store Size*: 2000

Crypto Core Capacity*: 8

SSL Core Contexts*: 10000

Create
Close

The summary page displays all the partitions that were created. Some partitions are assigned an instance while some are free partitions.



NetScaler SDX > System > HSM Administration > Partitions
Add
Edit
Delete

Total Keys	Available Keys	Total Crypto Cores	Available Crypto Cores	Total SSL Contexts	Available SSL Contexts
102,235	97,035	63	23	1,000,000	610,000

Name	Key Store Size	Crypto Core Capacity	SSL Core Contexts	Instance Name
Part-3	2000	8	10000	
Part-4	200	2	10000	
Partition-1234	100	4	20000	
Partition-12345	300	4	20000	
Partition-5	300	8	100000	
Part-6	200	8	200000	
Part-1	100	2	10000	NSVPX-1-10.217.202.35
Part-2	2000	4	20000	NSVPX-2-10.217.202.36

Provision a new instance or modify an existing instance and assign a partition

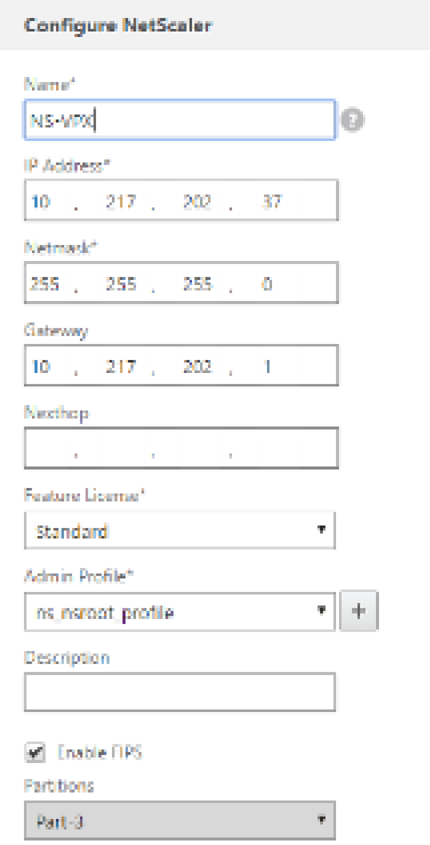
September 21, 2020

After creating the partitions, you must assign them to instances.

> **Important:**
>
> - You can attach only one FIPS partition to an instance.
> - An instance with a FIPS partition can be assigned only one CPU core.

Provision a new instance or modify an existing instance

1. On the Configuration tab, navigate to **NetScaler** > **Instances**, and add or modify an instance.
2. Select **Enable FIPS**, and from the **Partitions** list, select a partition to attach to this instance.

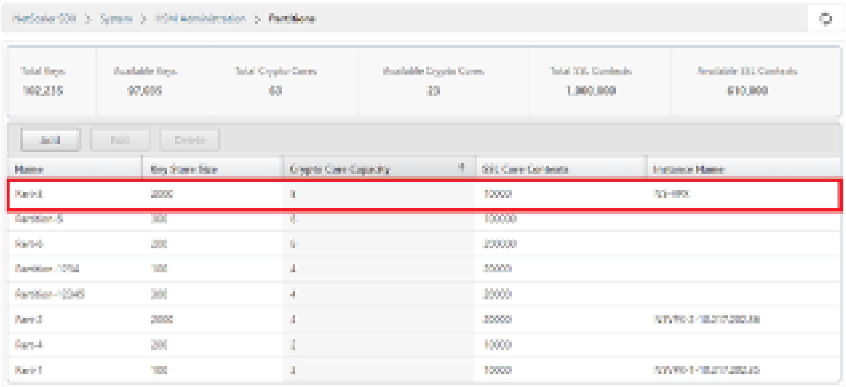


Configure NetScaler

Name*: NS-VPX
IP Address*: 10 . 217 . 202 . 37
Netmask*: 255 . 255 . 255 . 0
Gateway: 10 . 217 . 202 . 1
Nexthop: . . .
Feature License*: Standard
Admin Profile*: ns_nsroot_profile
Description:
☑ Enable FIPS
Partitions: Part-3

You can verify that the partition is attached to an instance by using either the GUI or the CLI.

In the GUI, navigate to **System** > **HSM Administration** > **Partitions**. The instance name attached to the partition is displayed.



To unassign a FIPS partition¸ navigate to **NetScaler** > **Instances**. Edit the instance and clear the **Enable FIPS** check box.

In the CLI, at the command prompt, type the following commands:

```
> show fips

FIPS Card is not configured

 Done
<!--NeedCopy-->
```

If you see the following output, see the troubleshooting section for debugging.

ERROR: Operation not permitted - no FIPS card present in the system

Configure the HSM for an instance on an SDX 14030/14060/14080 FIPS appliance

September 21, 2020

You must first check the state of your FIPS card to verify that the driver loaded correctly, and then initialize the card.

At the command prompt, type:

```
show fips

FIPS Card is not configured

Done
<!--NeedCopy-->
```

If the driver is not loaded correctly, the message "ERROR: Operation not permitted - no FIPS card present in the system" appears.

Initialize the FIPS card

> **Important:**
>
> Verify that the `/nsconfig/fips` directory has successfully been created on the appliance.

Do not save the configuration before you restart the appliance for the third time.

Perform the following steps to initialize the FIPS card:

1. Reset the FIPS card.
2. Restart the appliance.
3. Set the security officer password for partitions 0 and 1, and the user password for partition 1. Note: The set or reset command takes more than 60 seconds to run.
4. Save the configuration.
5. Verify that the password encrypted key for the master partition (master_pek.key) has been created in the /nsconfig/fips/ directory.
6. Restart the appliance.
7. Verify that the FIPS card is UP.

Initialize the FIPS card by using the CLI

At the command prompt, type the following commands:

```
reset fips

reboot

set fips -initHSM Level-2 <soPassword> <oldsoPassword> <userPassword> -hsmLabel <string>
<!--NeedCopy-->
```

> **Note**: The following message appears when you run the **set fips** command:

```
This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y

saveconfig

```

```
reboot

show fips
<!--NeedCopy-->
```

Example:

```
reset fips

Done

reboot

set fips -initHSM Level-2 so12345 so12345 user123 -hsmLabel NSFIPS

This command will erase all data on the FIPS card. You must save the configuration (saveconfig) after executing this command. [Note: On MPX/SDX 14xxx FIPS platform, the FIPS security is at Level-3 by default, and the -initHSM Level-2 option is internally converted to Level-3]  Do you want to continue?(Y/N)y


Done

saveconfig

Done

reboot

show fips

    FIPS HSM Info:

    HSM Label : NSFIPS

    Initialization : FIPS-140-2 Level-2

    HSM Serial Number : 3.0G1532-ICM000228

    HSM State : 2

    HSM Model : NITROX-III CNN35XX-NFBE

    Hardware Version : 0.0-G

    Firmware Version : 1.0

    Firmware Build : NFBE-FW-1.0-48

    Max FIPS Key Memory : 1000

    Free FIPS Key Memory : 1000

    Total SRAM Memory : 557396

    Free SRAM Memory : 238088

    Total Crypto Cores : 4

    Enabled Crypto Cores : 4

Done
<!--NeedCopy-->
```

Create a FIPS key for an instance on an SDX 14030/14060/14080 FIPS appliance

September 21, 2020

You can create a FIPS key on your instance or import an existing FIPS key into the instance. An SDX 14030/14060/14080 FIPS appliance supports only 2048-bit and 3072-bit keys and an exponent value of F4. For PEM keys, an exponent is not required. Verify that the FIPS key is created correctly. Create a certificate signing request and a server certificate. Finally, add the certificate-key pair to your instance.

> **Note:**
>
> 1024-bit and 4096-bit keys and an exponent value of 3 are not supported.

Create a FIPS key by using the CLI

At the command prompt, type:

```
create ssl fipsKey <fipsKeyName> -keytype ( RSA | ECDSA ) [-exponent (3 | F4 )] [-modulus <positive_integer>] [-curve ( P_256 | P_384 )]
<!--NeedCopy-->
```

Example:

```
create fipsKey f1 -keytype RSA -modulus 2048 -exponent F4

Done

show ssl fipskey ddvws

FIPS Key Name: f1  Key Type: RSA Modulus: 2048   Public Exponent: F4 (
    Hex: 0x10001)

Done
<!--NeedCopy-->
```

Import a FIPS key by using the CLI

At the command prompt, type:

```
import ssl fipsKey <fipsKeyName> -key <string> [-inform <inform>] [-
    wrapKeyName <string>] [-iv<string>] [-exponent F4 ]
<!--NeedCopy-->
```

Example:

```
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform SIM -exponent F4
Done
import fipskey Key-FIPS-2 -key Key-FIPS-2.key -inform PEM
Done
<!--NeedCopy-->
```

Verify that the FIPS key is created or imported correctly by running the **show fipskey** command.

```
show fipskey
1)       FIPS Key Name: Key-FIPS-2
Done
<!--NeedCopy-->
```

Create a certificate signing request by using the CLI

At the command prompt, type:

```
create ssl certReq <reqFile> (-keyFile <input_filename> | -fipsKeyName
    <string>) [-keyform ( DER | PEM )  {
 -PEMPassPhrase  }
 ] -countryName <string> -stateName <string> -organizationName<string> [-organizationUnitName <string>] [-localityName <string>] [-commonName <string>] [-emailAddress <string>] {
 -challengePassword  }
  [-companyName <string>] [-digestMethod ( SHA1 | SHA256 )]
<!--NeedCopy-->
```

Example:

```
create certreq f1.req  - fipsKeyName f1 -countryName US -stateName CA -organizationName Citrix -companyName Citrix -commonName ctx -emailAddress test@example.com`
`Done
<!--NeedCopy-->
```

Create a server certificate by using the CLI

At the command prompt, type:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform ( DER | PEM ) {
 -PEMPassPhrase  }
 ] [-days <positive_integer>] [-certForm ( DER | PEM )] [-CAcert <input_filename>] [-CAcertForm ( DER | PEM )] [-CAkey <input_filename>] [-CAkeyForm ( DER | PEM )] [-CAserial <output_filename>]
<!--NeedCopy-->
```

Example:

```
create cert f1.cert f1.req SRVR_CERT -CAcert ns-root.cert -CAkey ns-root.key -CAserial ns-root.srl -days 1000
Done
<!--NeedCopy-->
```

The above example creates a server certificate using a local root CA on the appliance.

Add a certificate-key pair by using the CLI

At the command prompt, type:

```
add ssl certKey <certkeyName> (-cert <string> [-password]) [-key <string> | -fipsKey <string> | -hsmKey <string>] [-inform <inform>]
```

```
    [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod   <
    positive_integer>]] [-bundle ( YES | NO )]
<!--NeedCopy-->
```

Example:

```
add certkey cert1 -cert f1.cert -fipsKey f1
Done
<!--NeedCopy-->
```

After creating the FIPS key and server certificate, you can add the generic SSL configuration. Enable the features that are required for your deployment. Add servers, services, and SSL virtual servers. Bind the certificate-key pair and the service to the SSL virtual server, and save the configuration.

```
enable ns feature SSL LB
Done
add server s1 10.217.2.5
Done
add service sr1 s1 HTTP 80
Done
add lb vserver v1 SSL 10.217.2.172 443
Done
bind ssl vserver v1  - certkeyName cert1
Done
bind lb vserver v1 sr1
Done
saveconfig
Done
<!--NeedCopy-->
```

For information about configuring secure HTTPS and secure RPC, click here.

Upgrade the FIPS firmware

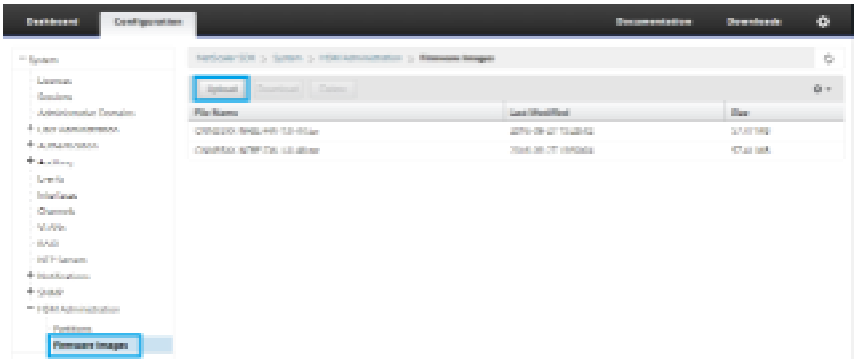
September 21, 2020

FIPS firmware updates are released from time to time. Download the latest firmware from the Citrix download page and upload it to the appliance. The upgrade process might take up to 10 minutes to complete. The instance is restarted after the upgrade.

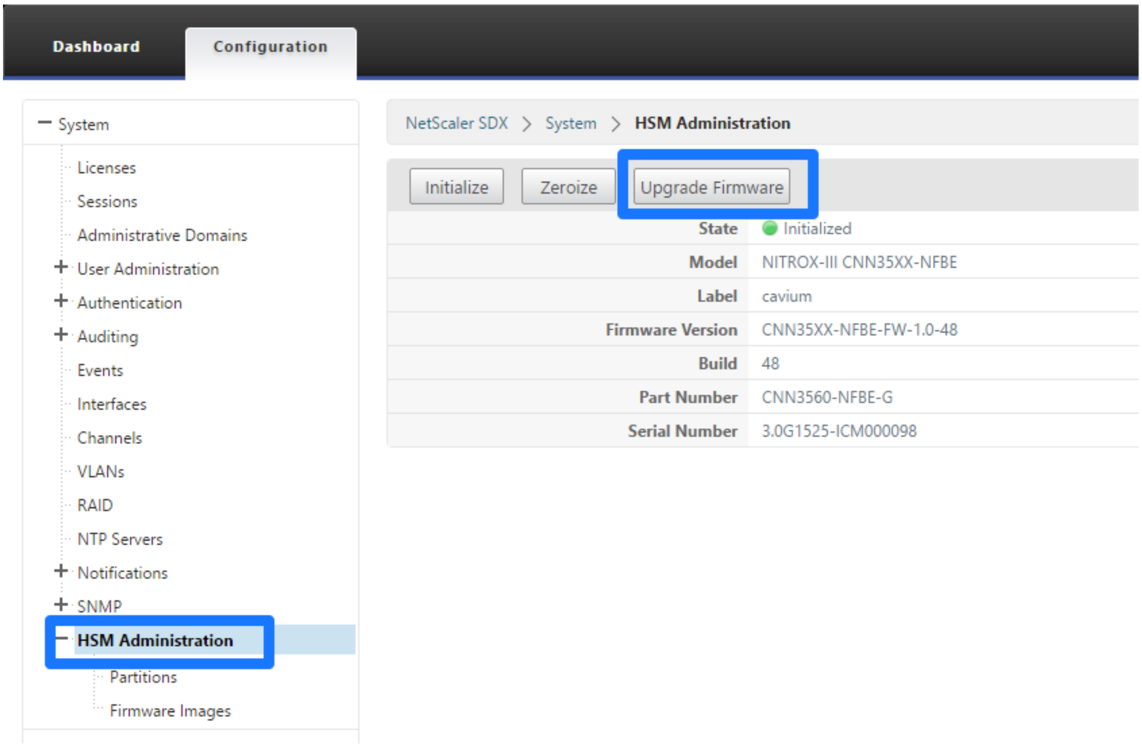
Upgrade the FIPS firmware

1. Navigate to **System** > **HSM Administration** > **Firmware Images**.

2. Select **Upload**.

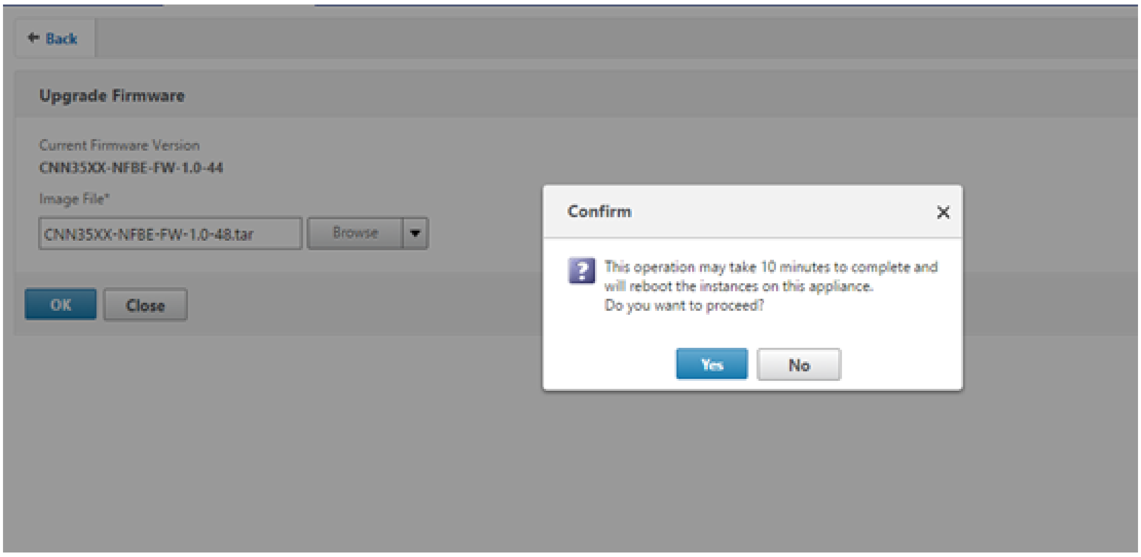


3. Navigate to the folder that contains the firmware image and select the file.
4. Navigate to **System** > **HSM Administration**, and select **Upgrade Firmware**.


Dashboard
Configuration
System
Licenses
Sessions
Administrative Domains
User Administration
Authentication
Auditing
Events
Interfaces
Channels
VLANs
RAID
NTP Servers
Notifications
SNMP
HSM Administration
Partitions
Firmware Images
NetScaler SDX > System > HSM Administration
Initialize
Zeroize
Upgrade Firmware

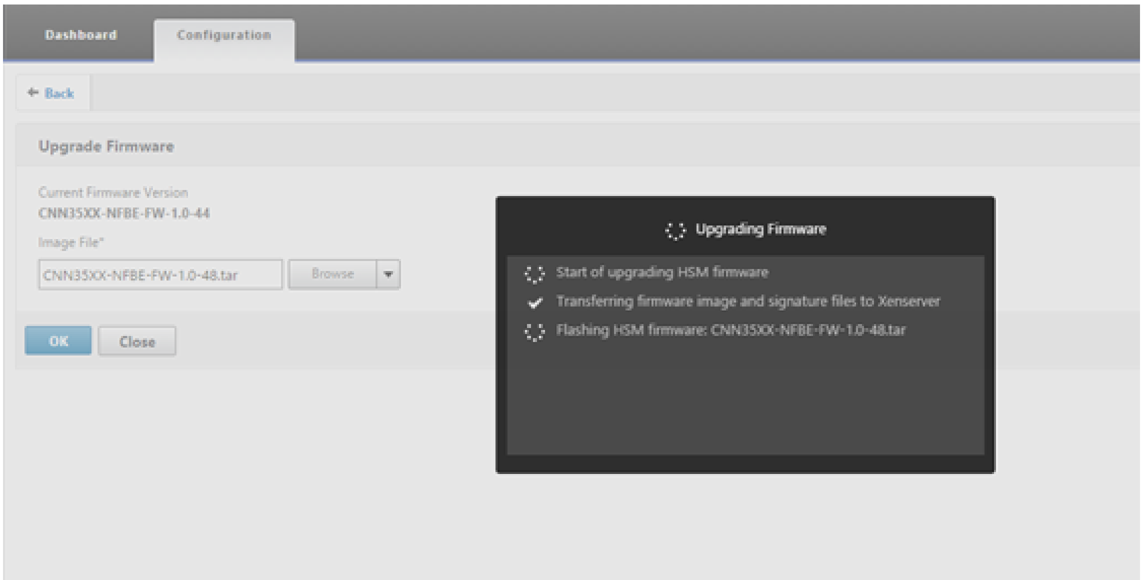
State	● Initialized
Model	NITROX-III CNN35XX-NFBE
Label	cavium
Firmware Version	CNN35XX-NFBE-FW-1.0-48
Build	48
Part Number	CNN3560-NFBE-G
Serial Number	3.0G1525-ICM000098

5. Select the firmware image to upgrade to, and click **OK**.


Back
Upgrade Firmware
Browse
OK
Close
Confirm
Yes
No

Current Firmware Version
CNN35XX-NFBE-FW-1.0-44
Image File*
CNN35XX-NFBE-FW-1.0-48.tar

This operation may take 10 minutes to complete and will reboot the instances on this appliance.
Do you want to proceed?


Dashboard
Configuration
Back
Upgrade Firmware
Browse
OK
Close
Upgrading Firmware

Current Firmware Version
CNN35XX-NFBE-FW-1.0-44
Image File*
CNN35XX-NFBE-FW-1.0-48.tar

Start of upgrading HSM firmware
✓ Transferring firmware image and signature files to Xenserver
Flashing HSM firmware: CNN35XX-NFBE-FW-1.0-48.tar

Support for Thales nShield® HSM

September 21, 2020

A non-FIPS NetScaler appliance stores the server’s private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as hardware security module (HSM). Storing a key in the HSM protects it from physical and software attacks. In addition, the keys are encrypted by using special FIPS approved ciphers.

Only the NetScaler MPX 9700/10500/12500/15500 FIPS appliances support a FIPS card. Support for FIPS is not available on other MPX appliances, or on the SDX and VPX appliances. This limitation is addressed by supporting a Thales nShield® Connect external HSM on all NetScaler MPX, SDX, and VPX appliances except the MPX 9700/10500/12500/15500 FIPS appliances.

Thales nShield Connect is an external FIPS-certified network-attached HSM. With a Thales HSM, the keys are securely stored as application key tokens on a remote file server (RFS) and can be reconstituted inside the Thales HSM only.

If you are already using a Thales HSM, you can now use a NetScaler to optimize, secure, and control the delivery of all enterprise and cloud services.

> **Note:**
>
> - Thales HSMs comply with FIPS 140-2 Level 3 specifications, while the MPX FIPS appliances comply with level 2 specifications.
> - You cannot decrypt the trace while using the Thales HSM, because the response from the HSM to the NetScaler appliance is encrypted and only the Hardserver can read it.

Supported versions matrix

Citrix ADC Version	Thales Client Version	Hardserver Version	Thales Firmware Version
10.5e, 11.0, 11.1, 12.0, 12.1	11.70, 11.72	2.71.2	2.50.16, 2.51.10

Architecture overview

September 21, 2020

The three entities that are part of a NetScaler-Thales deployment are a Thales nShield Connect module, a remote file server (RFS), and a NetScaler.

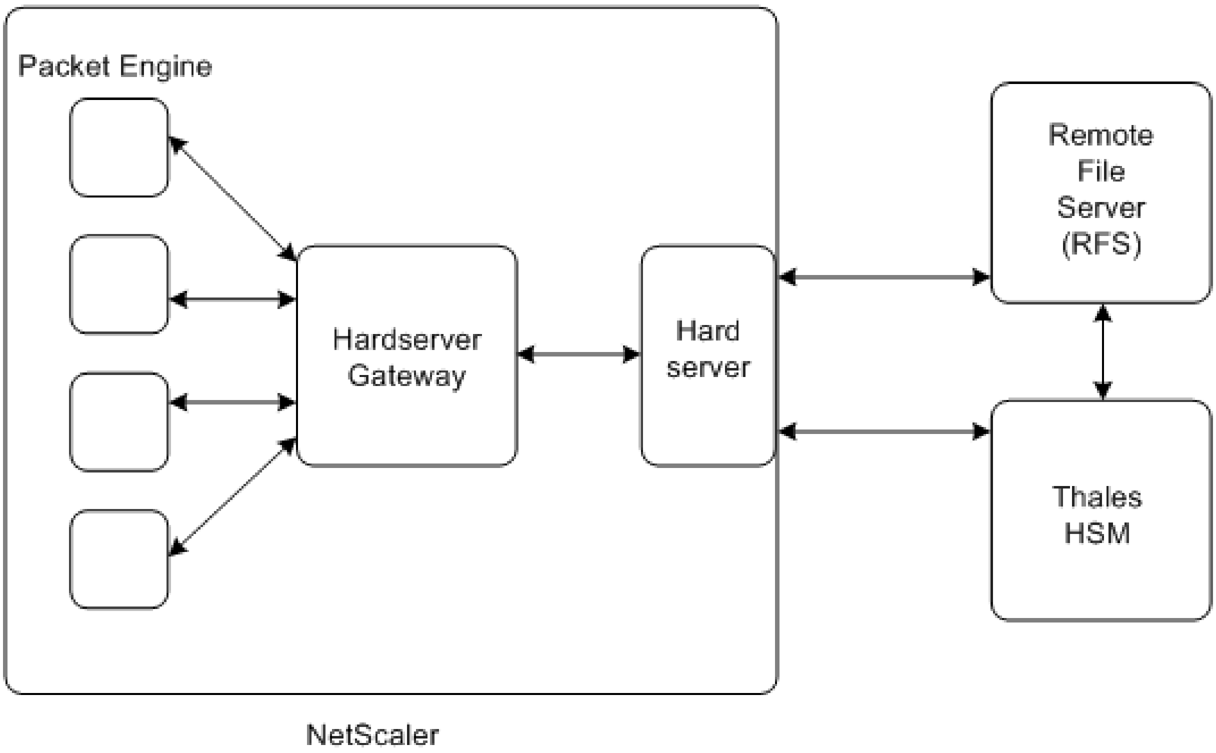
The Thales nShield Connect is a network-attached hardware security module. The RFS is used to configure the HSM and to store the encrypted key files.

Hardserver, a proprietary daemon provided by Thales, is used for communication between the client (ADC), the Thales HSM, and the RFS. It uses the IMPATH secure communication protocol. A gateway daemon, called the Hardserver Gateway, is used to communicate between the NetScaler packet engine and the Hardserver.

> **Note:**
>
> The terms Thales nShield Connect, Thales HSM, and HSM are used interchangeably in this docu-

> mentation.

The following figure illustrates the interaction between the different components.

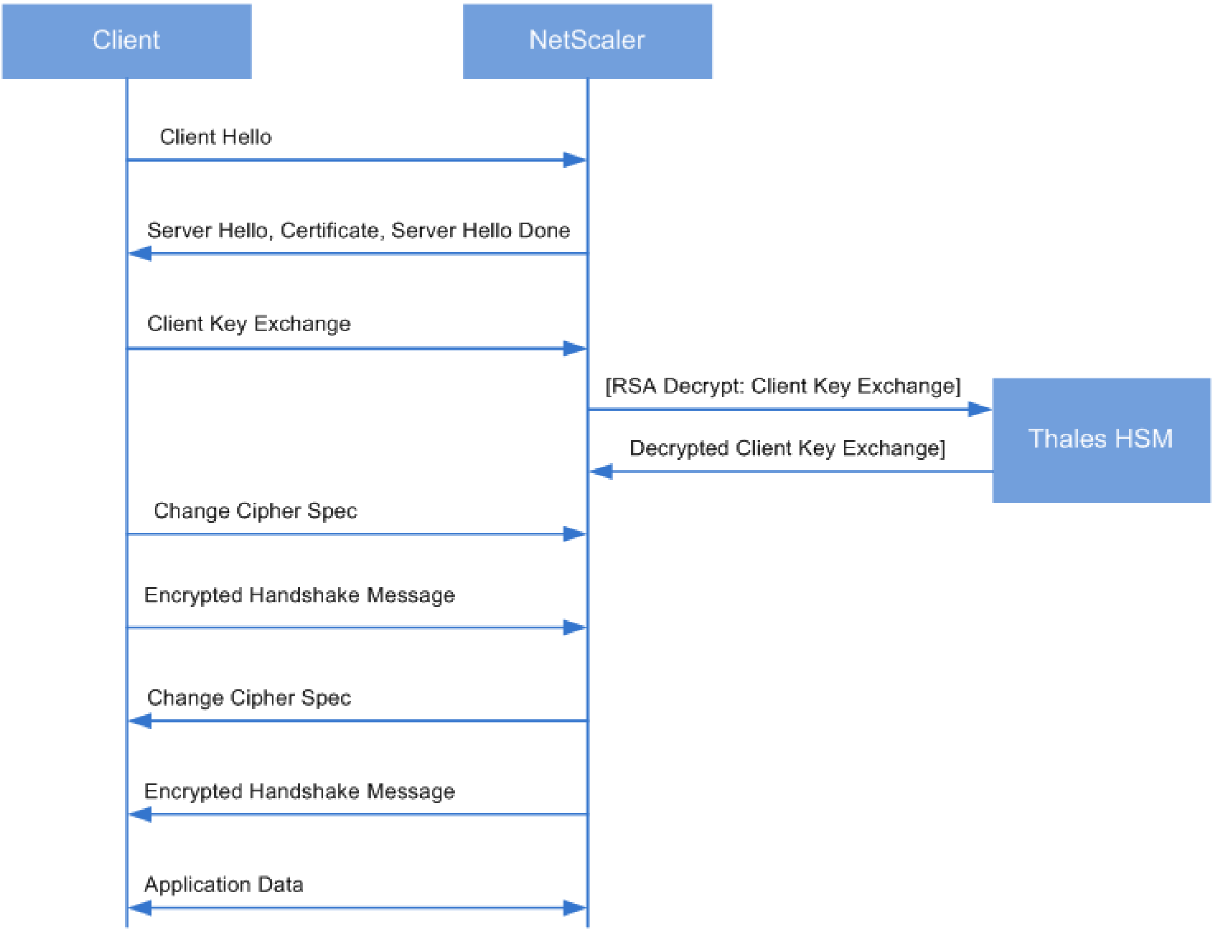

Packet Engine
Hardserver Gateway
Hard server
Remote File Server (RFS)
Thales HSM
NetScaler

In a typical deployment, the RFS is used to securely store keys generated by the HSM. After the keys are generated, you can securely transfer them to the ADC and then use the GUI or command line to load the keys to the HSM. A virtual server on the ADC uses Thales to decrypt the client key exchange to complete the SSL handshake. Thereafter, all the SSL operations are performed on the ADC.

> **Note:**
>
> The terms keys and application key tokens are used interchangeably in this documentation.

The following figure illustrates the packet flow in the SSL handshake with the Thales HSM.

Figure 1. SSL Handshake Packets Flow Diagram with NetScaler Using Thales HSM


Client
NetScaler
Client Hello
Server Hello, Certificate, Server Hello Done
Client Key Exchange
[RSA Decrypt: Client Key Exchange]
Decrypted Client Key Exchange]
Thales HSM
Change Cipher Spec
Encrypted Handshake Message
Change Cipher Spec
Encrypted Handshake Message
Application Data

> **Note:**
>
> The communication between the ADC and the HSM uses a Thales proprietary communication protocol, called IMPATH.

Prerequisites

September 21, 2020

Before you can use a Thales nShield Connect with a NetScaler, make sure that the following prerequisites are met:

- A Thales nShield Connect device is installed in the network, ready to use, and accessible to the NetScaler. That is, the NSIP address or the SNIP address is added as an authorized client on the HSM.
- A usable Security World exists. Security World is a unique key management architecture used by the Thales nShield line of HSMs. It protects and manages keys as application key tokens, enabling unlimited key capacity, and automatic key backup and recovery. For more information

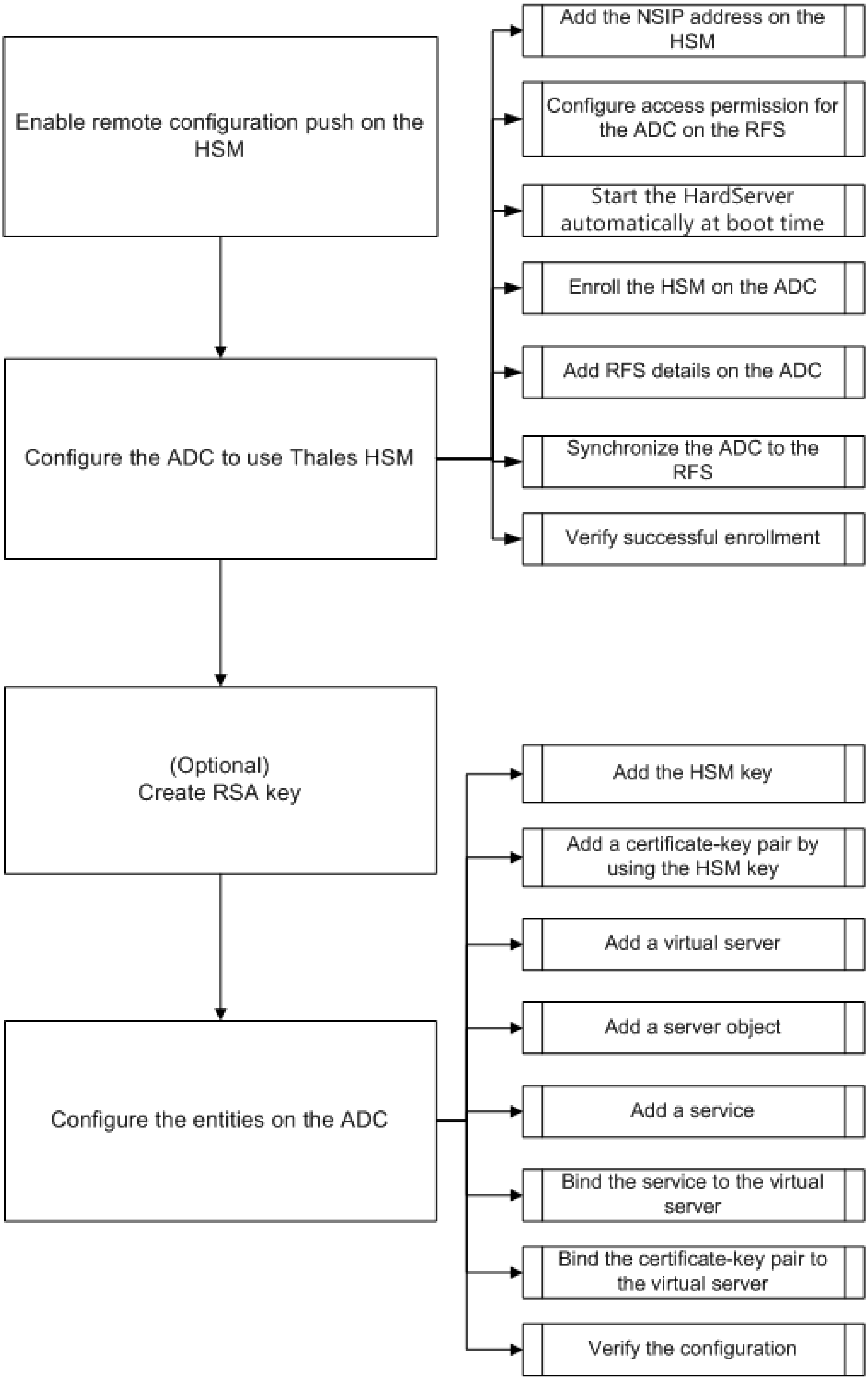
about creating a Security World, see the nShield Connect Quick Start Guide from Thales. You can also find the guide in the CD provided with the Thales HSM module at CipherTools-linux-dev-xx.xx.xx/document/nShield_Connect_Quick_Start_Guide.pdf.
Note: Softcard or token/OCS protected keys are currently not supported on the NetScaler.

- Licenses are available to support the number of clients that are connected to the Thales HSM. The ADC and remote file server (RFS) are clients of the HSM.
- An RFS is installed in the network and is accessible to the NetScaler.
- The Thales nShield Connect device, the RFS, and the NetScaler can initiate connections with each other through port 9004.
- You are using NetScaler release 10.5 build 52.1115.e or later.
- The NetScaler appliance does not contain a FIPS Cavium card.
  Important: Thales HSM is not supported on the MPX 9700/10500/12500/15500 FIPS appliances.

Configure the ADC-Thales integration

September 21, 2020

The following flowchart depicts the tasks that you need to perform to use Thales HSM with a NetScaler:


Enable remote configuration push on the HSM
Add the NSIP address on the HSM
Configure access permission for the ADC on the RFS
Start the HardServer automatically at boot time
Enroll the HSM on the ADC
Add RFS details on the ADC
Configure the ADC to use Thales HSM
Synchronize the ADC to the RFS
Verify successful enrollment
(Optional)
Create RSA key
Add the HSM key
Add a certificate-key pair by using the HSM key
Add a virtual server
Add a server object
Add a service
Configure the entities on the ADC
Bind the service to the virtual server
Bind the certificate-key pair to the virtual server
Verify the configuration

As shown in the above flowchart, you perform the following tasks:

1. Enable remote configuration push on the HSM.
2. Configure the ADC to use the Thales HSM.
    - Add the NSIP address on the HSM.
    - Configure access permission for the ADC on the RFS.
    - Configure automatic start of the Hardserver at boot time.
    - Enroll the HSM on the ADC.
    - Add RFS details on the ADC.
    - Synchronize the ADC to the RFS.
    - Verify that Thales HSM is successfully enrolled on the ADC.
3. (Optional) Create an HSM RSA key.
4. Configure the entities on the NetScaler.
    - Add the HSM key.
    - Add a certificate-key pair by using the HSM key.
    - Add a virtual server.
    - Add a server object.
    - Add a service.
    - Bind the service to the virtual server.
    - Bind the certificate-key pair to the virtual server.
    - Verify the configuration.

Configure the Thales HSM

You must specify the IP address of the RFS on the Thales HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Thales HSM to perform the following procedure.

Specify the IP address of a remote computer on the Thales HSM

1. Navigate to **System Configuration** > **Config file options** > **Allow auto push**.
2. Select **ON**, and specify the IP address of the computer (RFS) from which to accept the configuration.

Enable remote configuration push on the HSM

You must specify the IP address of the RFS on the Thales HSM so that it accepts the configuration that the RFS pushes to it. Use the nShield Connect front panel on the Thales HSM to perform the following procedure.

Specify the IP address of a remote computer on the Thales HSM

1. Navigate to System Configuration > Config file options > Allow auto push.
2. Select ON, and specify the IP address of the computer (RFS) from which to accept the configuration.

Configure the ADC to use the Thales HSM

Sample values used in this documentation:

NSIP address=10.217.2.43

Thales HSM IP address=10.217.2.112

RFS IP address=10.217.2.6

Add the NSIP address on the HSM

Typically you use the nShield Connect front panel to add clients to the HSM. For more information, see the nShield Connect Quick Start Guide.

Alternately, use the RFS to add the ADC as a client to the HSM. To do this, you must add the NSIP address in the HSM configuration on the RFS, and then push the configuration to the HSM. Before you can do this, you must know the electronic serial number (ESN) of the HSM.

To get the ESN of your HSM, run the following command on the RFS:

```
root@ns# /opt/nfast/bin/anonkneti <Thales HSM IP address>
<!--NeedCopy-->
```

Example:

```
root@ns# /opt/nfast/bin/anonkneti 10.217.2.112
BD17-C807-58D9 5e30a698f7bab3b2068ca90a9488dc4e6c78d822
<!--NeedCopy-->
```

The ESN number is BD17-C807-58D9.

After you have the ESN number, use an editor, such as vi, to edit the HSM configuration file on the RFS.

```
vi /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config`
<!--NeedCopy-->
```

In the hs_clients section, add the following entries:

```
# Amount of data in bytes to encrypt with a session key before session key# renegotiation, or 0 for unlimitied. (default=1024*1024*8b=8Mb).
#  datalimit=INT
addr=10.217.2.43
clientperm=unpriv
keyhash=0000000000000000000000000000000000000000
esn=
timelimit=86400
datalimit=8388608
-----
<!--NeedCopy-->
```

Note: Include one or more hyphens as delimiters to add multiple entries in the same section.

To push the configuration to the HSM, run the following command on the RFS:

```
/opt/nfast/bin/cfg-pushnethsm --address=<Thales HSM IP address> --force /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

Example:

```
/opt/nfast/bin/cfg-pushnethsm --address=10.217.2.112 --force
                /opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
<!--NeedCopy-->
```

Configure access permission for the ADC on the RFS

To configure access permission for the ADC on the RFS, run the following command on the RFS:

```
/opt/nfast/bin/rfs-setup --force -g --write-noauth <NetScaler IP address>
<!--NeedCopy-->
```

Example:

```
[root@localhost bin]# /opt/nfast/bin/rfs-setup --force -g --write-noauth 10.217.2.43
Adding read-only remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local exists
Adding new writable remote_file_system entries
Ensuring the directory /opt/nfast/kmdata/local/sync-store exists
Saving the new config file and configuring the hardserver
Done
<!--NeedCopy-->
```

Verify that the ADC can reach both the RFS and Thales HSM by using port 9004.

Configure automatic start of the hardserver at boot time

Create a file and then restart the appliance. Now, whenever you restart the appliance, and if this file is found, the Hardserver is automatically started.

At the shell prompt, type:

```
touch /var/opt/nfast/bin/thales_hsm_is_enrolled
<!--NeedCopy-->
```

At the command prompt, type:

```
reboot
<!--NeedCopy-->
```

Enroll the HSM on the ADC

Change directory to /var/opt/nfast/bin.

To add HSM details into the ADC configuration, run the following command on the ADC:

```
nethsmenroll --force <Thales_nShield_Connect_ip_address> $(anonkneti <
    Thales_nShield_Connect_ip_address>)
<!--NeedCopy-->
```

Example:

```
root@ns# ./nethsmenroll --force 10.217.2.112 $(anonkneti 10.217.2.112)
OK configuring hardserver's nethsm imports
<!--NeedCopy-->
```

This step adds the following entries after the line # ntoken_esn=ESN in the nethsm_imports section of the /var/opt/nfast/kmdata/config/config file.

```
…
local_module=0
remote_ip=10.217.2.112
remote_port=9004
remote_esn=BD17-C807-58D9
keyhash=5e30a698f7bab3b2068ca90a9488dc4e6c78d822
timelimit=86400
datalimit=8388608
privileged=0
privileged_use_high_port=0
ntoken_esn=
<!--NeedCopy-->
```

Change directory to /var/opt/nfast/bin and run the following command on the ADC:

```
touch "thales_hsm_is_enrolled"
<!--NeedCopy-->
```

Note: To remove an HSM that is enrolled on the ADC, type:

> ```
> ./nethsmenroll - -remove <NETHSM-IP>
> <!--NeedCopy-->
> ```

Add RFS details on the ADC

To add RFS details, change directory to /var/opt/nfast/bin/ and then run the following command:

```
./rfs-sync --no-authenticate --setup <rfs_ip_address>
<!--NeedCopy-->
```

Example:

```
./rfs-sync --no-authenticate --setup 10.217.2.6
No current RFS synchronization configuration.
Configuration successfully written; new config details:
Using RFS at 10.217.2.6:9004: not authenticating.
<!--NeedCopy-->
```

This step adds the following entries after the # local_esn=ESN line in the rfs_sync_client section of the /var/opt/nfast/kmdata/config/config file.

```
… …
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
<!--NeedCopy-->
```

Note: To remove an RFS that is enrolled on the ADC, type:

> ```
> ./rfs_sync  - remove
> <!--NeedCopy-->
> ```

Synchronize the ADC to the RFS

To synchronize all the files, change directory to /var/opt/nfast/bin and then run the following command on the ADC:

```
./rfs-sync - -update
<!--NeedCopy-->
```

This command fetches all the World files, module files, and key files from the /opt/nfast/kmdata/local directory on the RFS and puts them into the /var/opt/nfast/kmdata/local directory on the ADC. Citrix recommends that you manually copy the World files, the module_XXXX_XXXX_XXXX files, where XXXX_XXXX_XXXX is the ESN of the enrolled HSM, and only the required RSA key and certificate files.

Verify that the Thales HSM is successfully enrolled on the ADC

After you synchronize the ADC to the RFS, do the following:

- Verify that the local Hardserver is UP and running. (nCipher server running).
- Get the state of the configured HSMs, and verify that the values for the n_modules (number of modules) field and the km info fields are non-zero.
- Verify that the HSM is enrolled correctly and is usable (state 0x2 Usable) by the ADC.
- Load tests using sigtest run properly.

Change directory to
/var/opt/nfast/bin, and at the shell prompt, run the following commands:

```
root@ns# ./chkserv root@ns# ./nfkminfo root@ns# ./sigtest
<!--NeedCopy-->
```

See
Appendix for an example.

Create an HSM RSA Key

Only RSA keys are supported as HSM keys.

Note: Skip this step if keys are already present in the
/opt/nfast/kmdata/local folder on the RFS.

Create an RSA key, a self-signed certificate, and a Certificate Signing Request (CSR). Send the CSR to a certificate authority to get a server certificate.

The following files are created in the example below:

- Embed RSA key: key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
- Self-Signed Certificate: example_selfcert
- Certificate Signing Request: example_req

Note: The
generatekey command is supported in strict FIPS 140-2 Level 3 Security World. An administrator card

set (ACS) or an operator card set (OCS) is needed to control many operations, including the creation of keys and OCSs. When you run the
generatekey command, you are prompted to insert an ACS or OCS card. For more information about strict FIPS 140-2 Level 3 Security World, see the nShield Connect User Guide.

The following example uses Level-2 Security World. In the example, the commands are in boldface type.

Example:

```
[root@localhost bin]# ./generatekey embed
size: Key size? (bits, minimum 1024) [1024] > 2048
OPTIONAL: pubexp: Public exponent for RSA key (hex)? []
>
embedsavefile: Filename to write key to? []
> example
plainname: Key name? [] > example
x509country: Country code? [] > US
x509province: State or province? [] > CA
x509locality: City or locality? [] > Santa Clara
x509org: Organisation? [] > Citrix
x509orgunit: Organisation unit? [] > NS
x509dnscommon: Domain name? [] > www.citrix.com
x509email: Email address? [] > example@citrix.com
nvram: Blob in NVRAM (needs ACS)? (yes/no) [no] >
digest: Digest to sign cert req with? (md5, sha1, sha256, sha384, sha512)
  [default sha1] > sha512
key generation parameters:
 operation      Operation to perform               generate
 application    Application                        embed
 verify         Verify security of key             yes
 type           Key type                           RSA
 size           Key size                           2048
 pubexp         Public exponent for RSA key (hex)
 embedsavefile  Filename to write key to           example
 plainname      Key name                           example
 x509country    Country code                       US
 x509province   State or province                  CA
 x509locality   City or locality                   Santa Clara
 x509org        Organisation                       Citrix
 x509orgunit    Organisation unit                  NS
 x509dnscommon  Domain name                        www.citrix.com
 x509email      Email address                      example@citrix.com
 nvram          Blob in NVRAM (needs ACS)          no
 digest          Digest to sign cert req with       sha512
Key successfully generated.
Path to key: /opt/nfast/kmdata/local/
    key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2c78
You have new mail in /var/spool/mail/root
<!--NeedCopy-->
```

Result:

You have created a CSR (example_req), a self-signed certificate (example_selfcert), and an application key token file in embed format (/opt/nfast/kmdata/local/key_embed_2ed5428aaeae1e159bdbd63f25292c7113ec2

Because the ADC supports keys in simple format only, you must convert the embed key to a simple key.

To convert the embed key to a simple key, run the following command on the RFS:

```
[root@localhost bin]# ./generatekey -r simple
from-application: Source application? (embed, simple) [embed] > embed
from-ident: Source key identifier? (
    c6410ca00af7e394157518cb53b2db46ff18ce29,
                                     2
                                         ed5428aaeae1e159bdbd63f25292c7113ec2c78
                                         )
  [default c6410ca00af7e394157518cb53b2db46ff18ce29]
> 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
ident: Key identifier? [] > examplersa2048key
plainname: Key name? [] > examplersa2048key
key generation parameters:
 operation          Operation to perform    retarget
 application        Application             simple
 verify             Verify security of key  yes
 from-application   Source application      embed
 from-ident         Source key identifier   2
     ed5428aaeae1e159bdbd63f25292c7113ec2c78
 ident              Key identifier          examplersa2048key
 plainname          Key name                examplersa2048key
Key successfully retargetted.
Path to key: /opt/nfast/kmdata/local/key_simple_examplersa2048key
<!--NeedCopy-->
```

> **Important:**
>
> When prompted for the source key identifier, enter **2ed5428aaeae1e159bdbd63f25292c7113ec2c78** as the embed key.

Result:

A key with the prefix key_simple (for example key_simple_examplersa2048key) is created.

Note: examplersa2048key is the key identifier (ident) and is referred to as the HSM key name on the ADC. A key identifier is unique. All the simple files have the prefix key_simple.

Configure the Entities on the ADC

Before the ADC can process traffic, you must do the following:

1. Enable features.
2. Add a subnet IP (SNIP) address.
3. Add the HSM key to the ADC.
4. Add a certificate-key pair by using the HSM key.
5. Add a virtual server.
6. Add a server object.
7. Add a service.
8. Bind the service to the virtual server.
9. Bind the certificate-key pair to the virtual server.
10. Verify the configuration.

Enable features on the ADC

Licenses must be present on the ADC before you can enable a feature.

Enable a feature by using the CLI

At the command prompt, run the following commands:

- enable feature lb
- enable feature ssl

Enable a feature by using the GUI

Navigate to **System** > **Settings** and, in the **Modes and Features** group, select **Configure basic features••, and then select ••SSL Offloading**.

Add a subnet IP address

For more information about subnet IP addresses, see
Configuring Subnet IP Addresses.

Add a SNIP address and verify the configuration by using the CLI

At the command prompt, run the following commands:

```
add ns ip <IPAddress> <netmask> -type SNIP
show ns ip
<!--NeedCopy-->
```

Example:

```
> add ns ip 192.168.17.253 255.255.248.0 -type SNIP
Done
> show ns ip
        Ipaddress        Traffic Domain  Type             Mode     Arp
                Icmp     Vserver  State
        ---------        --------------  ----             ----     ---
                ----     -------  ------
1)      192.168.17.251   0               NetScaler IP     Active
   Enabled  Enabled  NA       Enabled
2)      192.168.17.252   0               VIP              Active
   Enabled  Enabled  Enabled  Enabled
3)      192.168.17.253   0               SNIP              Active
   Enabled  Enabled  NA       Enabled
 Done
<!--NeedCopy-->
```

Add a SNIP address and verify the configuration by using the GUI

Navigate to **System** > **Network** > **IPs**, add an IP address, and select the **IP Type** as Subnet IP.

Copy the HSM key and certificate to the ADC

Use a secure file transfer utility to securely copy the key (key_simple_examplersa2048key) to the /var/opt/nfast/kmdata/local folder, and the certificate (example_selfcert) to the /nsconfig/ssl folder on the ADC.

Add the key on the ADC

All the keys have a key-simple prefix. When adding the key to the ADC, use the ident as the HSM key name. For example, if the key that you added is key_simple_XXXX, the HSM key name is XXXX.

Important:

- The HSM key name must be the same as the ident that you specified when you converted an embed key to a simple key format.

- The keys must be present in the /var/opt/nfast/kmdata/local/ directory on the ADC.

Add an HSM key by using the CLI

At the shell prompt, run the following command:

```
add ssl hsmKey <hsmKeyName> -key <string>
<!--NeedCopy-->
```

Example:

```
> add ssl hsmKey examplersa2048key  - key key_simple_examplersa2048key
Done
<!--NeedCopy-->
```

Add an HSM key by using the GUI

Navigate to **Traffic Management** > **SSL** > **HSM**, and add an HSM key.

Add a certificate-key pair on the ADC

For information about certificate-key pairs, see
Add or update a certificate-key pair.

Add a certificate-key pair by using the CLI

At the command prompt, run the following command:

```
add ssl certKey <certkeyName> -cert <string> -hsmKey <string>
<!--NeedCopy-->
```

Example:

```
> add ssl certKey key22 -cert example_selfcert -hsmKey examplersa2048key
Done
<!--NeedCopy-->
```

Add a certificate-key pair by using the GUI

Navigate to **Traffic Management** > **SSL** > **Certificates**, and add a certificate-key pair.

Add a virtual server

For information about a virtual server, see SSL virtual server configuration.

Configure an SSL-based virtual server by using the CLI

At the command prompt, run the following command:

```
add lb vserver <name> <serviceType> <IPAddress> <port>
<!--NeedCopy-->
```

Example:

```
add lb vserver v1 SSL 192.168.17.252 443
<!--NeedCopy-->
```

Configure an SSL-based virtual server by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, create a virtual server, and specify the protocol as SSL.

Add a server object

Before you can add a server object on the ADC, make sure that you have created a backend server. The following example uses the built-in python HTTP Server module on a Linux system.

Example:

```
%python  - m SimpleHTTPServer 80
<!--NeedCopy-->
```

Add a server object by using the CLI

At the command prompt, run the following command:

```
add server <name> <IPAddress>
<!--NeedCopy-->
```

Example:

```
add server s1 192.168.17.246
<!--NeedCopy-->
```

Add a server object by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Servers**, and add a server.

Add a service

For more information, see
Configuring services.

Configure a service by using the CLI

At the command prompt, run the following command:

```
add service <name> <serverName> <serviceType> <port>
<!--NeedCopy-->
```

Example:

```
add service sr1 s1 HTTP 80
<!--NeedCopy-->
```

Configure a service by using the GUI

Navigate to **Traffic Management** > **Load Balancing** > **Services**, and create a service.

Bind the service to the virtual server

For more information, see
Bind services to the SSL virtual server.

Bind a service to a virtual server by using the CLI

At the command prompt, run the following command:

```
bind lb vserver <name> <serviceName>
<!--NeedCopy-->
```

Example:

```
bind lb vserver v1 sr1
<!--NeedCopy-->
```

Bind a service to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open a virtual server, and click in the Services pane to bind a service to the virtual server.

Bind the certificate-key pair to the virtual server on the ADC

For more information, see
Bind the certificate-key pair to the SSL virtual server.

Bind a certificate-key pair to a virtual server by using the CLI

At the command prompt, run the following command:

```
bind ssl vserver <vServerName> -certkeyName <string>
<!--NeedCopy-->
```

Example:

```
bind ssl vserver v1 -certkeyName key22
Warning: Current certificate replaces the previous binding
<!--NeedCopy-->
```

Bind a certificate-key pair to a virtual server by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**.
2. Open an SSL virtual server and, in **Advanced Settings**, click **SSL Certificate**.
3. Bind a server certificate to the virtual server.

Verify the configuration

To view the configuration by using the CLI:

At the command prompt, run the following commands:

```
show lb vserver <name>
show ssl vserver <vServerName>
<!--NeedCopy-->
```

Example:

```
show lb vserver v1
        v1 (192.168.17.252:443) - SSL   Type: ADDRESS
        State: UP
        Last state change was at Wed Oct 29 03:11:11 2014
        Time since last state change: 0 days, 00:01:25.220
        Effective State: UP
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Appflow logging: ENABLED
        No. of Bound Services :  1 (Total)       1 (Active)
        Configured Method: LEASTCONNECTION
        Current Method: Round Robin, Reason: Bound service's state changed to UP
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none
        L2Conn: OFF
        Skip Persistency: None
        IcmpResponse: PASSIVE
        RHIstate: PASSIVE
        New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
        Mac mode Retain Vlan: DISABLED
        DBS_LB: DISABLED
        Process Local: DISABLED
        Traffic Domain: 0

1) sr1 (192.168.17.246: 80) - HTTP State: UP    Weight: 1
Done
<!--NeedCopy-->
```

```
sh ssl vserver v1
        Advanced SSL configuration for VServer v1:
        DH: DISABLED
        Ephemeral RSA: ENABLED          Refresh Count: 0
        Session Reuse: ENABLED          Timeout: 120 seconds
        Cipher Redirect: DISABLED
        SSLv2 Redirect: DISABLED
        ClearText Port: 0
        Client Auth: DISABLED
        SSL Redirect: DISABLED
        Non FIPS Ciphers: DISABLED
        SNI: DISABLED
        SSLv2: DISABLED  SSLv3: DISABLED  TLSv1.0: ENABLED  TLSv1.1:
           DISABLED  TLSv1.2: DISABLED
        Push Encryption Trigger: Always
        Send Close-Notify: YES

        ECC Curve: P_256, P_384, P_224, P_521

1)      CertKey Name: key22      Server Certificate

1)      Cipher Name: DEFAULT
        Description: Predefined Cipher Alias
 Done
<!--NeedCopy-->
```

To view the configuration by using the GUI:

Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, and double-click an SSL virtual server to open it and view the configuration.

Limitations

September 21, 2020

- SSL version 3 (SSLv3) is not supported on an MPX appliance but is supported on a VPX virtual appliance. A VPX instance provisioned on an SDX appliance supports SSLv3 only if an SSL chip is not assigned to the instance.
- Export ciphers are not supported.
- SSL server key exchange using HSM keys is not supported.
- If you have added or removed keys after you last saved the configuration, you must save the configuration before you perform a warm restart. If you do not save the configuration, there will be a key mismatch between the ADC and the HSM.
- You cannot bind an HSM key to a DTLS virtual server.
- You cannot bind a certificate-key pair that is created by using an HSM key to an SSL service.
- You cannot use the GUI to enroll the ADC as a client of the HSM or check the status of the HSM from the configuration utility.
- From release 11 build 62.x, SSL renegotiation is supported.
- You cannot sign OCSP requests by using a certificate-key pair that is created by using an HSM key.
- A certificate bundle with HSM keys is not supported.
- An error does not appear if the HSM key and certificate do not match. Therefore, while adding a certificate-key pair, you need to make sure that the HSM key and certificate match.

- Clustering and admin partitions are not supported.

Appendix

September 21, 2020

Example:

Note: In the following example, the commands are in boldface type.

```
root@ns# ./chkserv
nCipher server running
root@ns# ./nfkminfo
World
 generation  2
 state       0x17a70000 Initialised Usable Recovery PINRecovery !
    ExistingClient RTC NVRAM FTO !AlwaysUseStrongPrimes SEEDebug
 n_modules   1
 hknso       cbec8c0c56c6b5e76b73147ef02d34a661eaa044
 hkm     bbb8d4839da5782be4d092735a7535538834dc91 (type Rijndael)
 hkmwk       1d572201be533ebc89f30fdd8f3fac6ca3395bf0
 hkre        01f21ecf43933ffdd45e74c3883525176c5c439c
 hkra        ac8ec5ee6bce00991bd97adce2091d9739b9b452
 hkmc        cf1b509abaad91995ed202d8f36613fc99433155
 hkp         c20910b2ed1ca62d6a2b0db67052a05f7bbfeb43
 hkrtc       bd811020a7c2f8df435a481c3767a89c2e13bc4f
 hknv        278b8012e48910d518a9ee91cff57233fb0c9093
 hkdsee      12230b0e31e3cec66324c0815f782cfb9249edd5
 hkfto       89dd6250b3d6149bcd15606f4553085e2fd6271a
 hkmnull     0100000000000000000000000000000000000000
 ex.client   none
 k-out-of-n  1/2
 other quora m=1 r=1 p=1 nv=1 rtc=1 dsee=1 fto=1
 createtime  2014-02-28 21:05:32
 nso timeout 10 min
 ciphersuite DLf1024s160mRijndael

Module #1
 generation 2
 state      0x2 Usable
 flags      0x10000 ShareTarget
 n_slots    2
 esn        BD17-C807-58D9
 hkml       70289a6edba00ddc7e3f6d6f5a49edc963e822f2
```

```

Module #1 Slot #0 IC 0
 generation    1
 phystype      SmartCard
 slotlistflags 0x2 SupportsAuthentication
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

Module #1 Slot #1 IC 0
 generation    1
 phystype      SoftToken
 slotlistflags 0x0
 state         0x2 Empty
 flags         0x0
 shareno       0
 shares
 error         OK
No Cardset

No Pre-Loaded Objects

root@ns# ./sigtest
 Hardware module #1 speed index 5792 recommended minimum queue 19
Found 1 module; using 19 jobs
Making 1024-bit RSAPrivate key on module #1;
  using Mech_RSApPKCS1 and PlainTextType_Bignum.
Generated and exported key from module #1.
Imported keys on module #1
 1,       3059 1223.6, 3059 overall
 2,       8698 2989.76, 4349 overall
 3,      14396 4073.06, 4798.67 overall
 4,      20091 4721.83, 5022.75 overall
 5,      25799 5116.3, 5159.8 overall
 6,      31496 5348.58, 5249.33 overall
 7,      37192 5487.55, 5313.14 overall
 8,      42780 5527.73, 5347.5 overall
 9,      45777 4515.44, 5086.33 overall
 10,      51457 4981.26, 5145.7 overall
 11,      57151 5266.36, 5195.55 overall
 12,      62813 5424.61, 5234.42 overall
 13,      68496 5527.97, 5268.92 overall
```

```
14,       74182 5591.18, 5298.71 overall
15,       79832 5614.71, 5322.13 overall
16,       85518 5643.23, 5344.88 overall
17,       88412 4543.54, 5200.71 overall
18,       94086 4995.72, 5227 overall
19,       99778 5274.23, 5251.47 overall
20,      105469 5440.94, 5273.45 overall
21,      111133 5530.16, 5292.05 overall
22,      116838 5600.1, 5310.82 overall
23,      122522 5633.66, 5327.04 overall
24,      128175 5641.4, 5340.62 overall
25,      131072 4543.64, 5242.88 overall
26,      136762 5002.18, 5260.08 overall
27,      142415 5262.51, 5274.63 overall
28,      148125 5441.51, 5290.18 overall
29,      153816 5541.3, 5304 overall
30,      159414 5563.98, 5313.8 overall
<!--NeedCopy-->
```

Support for Gemalto SafeNet Network hardware security module

September 21, 2020

A non-FIPS NetScaler appliance stores the server's private key on the hard disk. On a FIPS appliance, the key is stored in a cryptographic module known as a hardware security module (HSM). Storing a key in the HSM protects it from physical and software attacks. In addition, the keys are encrypted with special FIPS approved ciphers.

Only the NetScaler MPX 9700/10500/12500/15500 FIPS appliances and the MPX/SDX 14000 FIPS appliances support a FIPS card. Support for FIPS is not available on other MPX/SDX appliances, or on the NetScaler VPX appliances. This limitation is addressed by supporting a SafeNet network HSM on all NetScaler MPX, SDX, and VPX appliances except the MPX 9700/10500/12500/15500 FIPS and the MPX/SDX 14000 FIPS appliances.

A SafeNet network HSM is designed to protect critical cryptographic keys and to accelerate sensitive cryptographic operations across a wide range of security applications.

Supported versions matrix

NetScaler Version	Software Appliance Version	Firmware Version	Client Version
11.1, 12.0	5.2.3-1	6.2.1	6.0.0
11.1, 12.0	6.2.2-5	6.10.9	6.2.2

Prerequisites

September 21, 2020

Before you can use a SafeNet network HSM with a NetScaler, make sure that the following prerequisites are met:

- A SafeNet network HSM is installed in the network, ready to use, and accessible to the NetScaler. That is, the NSIP address is added as an authorized client on the HSM.
- Licenses are available to support the required number of partitions on the HSM.
- The SafeNet network HSM and the NetScaler can initiate connections with each other through port 1792.
- You are using NetScaler release 11.1 or later.
- The NetScaler appliance does not contain a FIPS Cavium card.

> **Important**
>
> SafeNet network HSMs are not supported on the MPX 9700/10500/12500/15500 FIPS appliances.

Configure a SafeNet client on the ADC

September 21, 2020

After you have configured the SafeNet HSM and created the required partitions, you must create clients and assign them to partitions. Begin by configuring the SafeNet clients on the NetScaler and setting up the network trust links (NTLs) between the SafeNet clients and the SafeNet HSM. A sample configuration is given in the Appendix.

1. Change directory to /var/safenet and install the Safenet client. At the shell prompt, type:

```
cd /var/safenet
<!--NeedCopy-->
```

To install Safenet client version 6.0.0, type:

```
install_client.sh -v 600
<!--NeedCopy-->
```

To install Safenet client version 6.2.2, type:

```
install\_client.sh -v 622
<!--NeedCopy-->
```

> **Note**
>
> SafeNet client version 6.2.2 is supported in release 12.0 build 51.x and later.

2. Configure the NTLs between SafeNet client (ADC) and HSM.

After the "/var/safenet/" directory is created, perform the following tasks on the ADC.

a) Change directory to /var/safenet/config/ and run the "safenet_config" script. At the shell prompt, type:

```
cd /var/safenet/config

sh safenet_config
<!--NeedCopy-->
```

This script copies the "Chrystoki.conf" file into the /etc/ directory. It also generates a symbolic link "libCryptoki2_64.so" in the "/usr/lib/" directory.

b) Create and transfer a certificate and key between the ADC and the SafeNet HSM.

In order to communicate securely, the ADC and the HSM must exchange certificates. Create a certificate and key on the ADC and then transfer it to the HSM. Copy the HSM certificate to the ADC.

i) Change directory to /var/safenet/safenet/lunaclient/bin.

ii) Create a certificate on the ADC. At the shell prompt, type:

```
./vtl createCert -n <ip address of NetScaler>
<!--NeedCopy-->
```

This command also adds the certificate and key path to the "/etc/Chrystoki.conf" file.

iii) Copy this certificate to the HSM. At the shell prompt, type:

```
scp /var/safenet/safenet/lunaclient/cert/client/<ip address of NS>.pem
   <LunaSA_HSM account>@<IP address of Luna SA>
<!--NeedCopy-->
```

iv) Copy the HSM certificate to the NetScaler. At the shell prompt, type:

```
scp <HSM account>@<HSM IP>:server.pem  /var/safenet/safenet/lunaclient/
    server_<HSM ip>.pem
<!--NeedCopy-->
```

3) Register the NetScaler as a client and assign it a partition on the SafeNet HSM.

Log on to the HSM and create a client. Enter the NSIP as the client IP. This must be the IP address of the ADC from which you transferred the certificate to the HSM. After the client is successfully registered, assign a partition to it. Run the following commands on the HSM.

a) Use SSH to connect to the SafeNet HSM and enter the password.

b) Register the NetScaler on the SafeNet HSM. The client is created on the HSM. The IP address is the client's IP address. That is, the NSIP address.

At the prompt, type:

```
client register  – client <client name> -ip <NetScaler ip>
<!--NeedCopy-->
```

c) Assign the client a partition from the partition list. To view the available partitions, type:

```
<luna_sh> partition list
<!--NeedCopy-->
```

Assign a partition from this list. Type:

```
<lunash:> client assignPartition -client <Client Name> -par <Partition
    Name>
<!--NeedCopy-->
```

4) Register the HSM with its certificate on the NetScaler.

On the ADC, change directory to "/var/safenet/safenet/lunaclient/bin" and, at the shell prompt, type:

```
./vtl addserver -n <IP addr of HSM> -c /var/safenet/safenet/lunaclient/
    server_<HSM_IP>.pem
<!--NeedCopy-->
```

To remove the HSM that is enrolled on the ADC, type:

```
./vtl deleteServer -n <HSM IP> -c <cert path>
<!--NeedCopy-->
```

To list the HSM servers configured on the ADC, type:

```
./vtl listServer
<!--NeedCopy-->
```

> **Note:**
>
> Before removing the HSM by using vtl, make sure all the keys for that HSM are manually removed from the appliance. HSM keys cannot be deleted after the HSM server is removed.

5) Verify the network trust links (NTLs) connectivity between the ADC and HSM. At the shell prompt, type:

```
./vtl verify
<!--NeedCopy-->
```

If verification fails, review all the steps. Errors are generally due to an incorrect IP address in the client certificates.

6) Save the configuration.

The above steps update the "/etc/Chrystoki.conf" configuration file. This file is deleted when the ADC is started. Copy the configuration to the default configuration file, which is used when an ADC is restarted.

At the shell prompt, type:

```
root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

Recommended practice is to run this command every time there is a change to the SafeNet-related configuration.

7) Start the SafeNet gateway process.

At the shell prompt, type:

```
sh /var/safenet/gateway/start_safenet_gw
<!--NeedCopy-->
```

8) Configure automatic start of the gateway daemon at boot time.

Create the "safenet_is_enrolled" file, which indicates that SafeNet HSM is configured on this ADC. Whenever the ADC restarts and this file is found, the gateway is automatically started.

At the shell prompt, type:

```
touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

Configure Safenet HSMs in a high availability setup on the ADC

September 21, 2020

Configuring SafeNet HSMs in a high availability (HA) ensures uninterrupted service even if all but one of the devices are unavailable. In an HA setup, each HSM joins an HA group in active-active mode. SafeNet HSMs in an HA setup provide load balancing of all the group members to increase performance and response time while providing the assurance of high availability service. For more information, contact SafeNet Sales and Support.

Prerequisites:

- Minimum two SafeNet HSM devices. All the devices in an HA group must have either PED (trusted path) authentication or password authentication. A combination of trusted path authentication and password authentication in an HA group is not supported.
- Partitions on each HSM device must have the same password even if the label (name) is different.
- All partitions in HA must be assigned to the client (NetScaler appliance).

After configuring a SafeNet client on the ADC as described in Configure a SafeNet client on the ADC, perform the following steps to configure Safenet HSMs in HA:

1. On the NetScaler shell prompt, launch "lunacm" (/usr/safenet/lunaclient/bin)

Example:

```
root@ns# cd /var/safenet/safenet/lunaclient/bin/

root@ns# ./lunacm
<!--NeedCopy-->
```

2. Identify the slot IDs of the partitions. To list the available slots (partitions), type:

```
lunacm:> slot list
<!--NeedCopy-->
```

Example;

```
        Slot Id ->              0


        HSM Label ->            trinity-p1

        HSM Serial Number ->    481681014

        HSM Model ->            LunaSA 6.2.1

        HSM Firmware Version -> 6.10.9

        HSM Configuration ->    Luna SA Slot (PED) Signing With Cloning Mode

        HSM Status ->           OK




        Slot Id ->              1

        HSM Label ->            trinity-p2

        HSM Serial Number ->    481681018

        HSM Model ->            LunaSA 6.2.1

        HSM Firmware Version -> 6.10.9

        HSM Configuration ->    Luna SA Slot (PED) Signing With Cloning Mode

        HSM Status ->           OK



         Slot Id ->              2

         HSM Label ->            neo-p1

         HSM Serial Number ->    487298014

         HSM Model ->            LunaSA 6.2.1

         HSM Firmware Version -> 6.10.9

         HSM Configuration ->    Luna SA Slot (PED) Signing With Cloning Mode

        HSM Status ->           OK



        Slot Id ->              3

        HSM Label ->            neo-p2

        HSM Serial Number ->    487298018

        HSM Model ->            LunaSA 6.2.1

        HSM Firmware Version -> 6.10.9

        HSM Configuration ->    Luna SA Slot (PED) Signing With Cloning Mode

        HSM Status ->           OK



        Slot Id ->              7

        HSM Label ->            hsmha

        HSM Serial Number ->    1481681014

        HSM Model ->            LunaVirtual

        HSM Firmware Version -> 6.10.9

        HSM Configuration ->    Luna Virtual HSM (PED) Signing With Cloning Mode

        HSM Status ->           N/A - HA Group


        Slot Id ->              8

        HSM Label ->            newha

        HSM Serial Number ->    1481681018

        HSM Model ->            LunaVirtual

        HSM Firmware Version -> 6.10.9

        HSM Configuration ->    Luna Virtual HSM (PED) Signing With Cloning Mode

        HSM Status ->            N/A - HA Group



        Current Slot Id: 0
<!--NeedCopy-->
```

3. Create the HA group. The first partition is called the primary partition. You can add more than one secondary partitions.

```
lunacm:> hagroup createGroup -slot <slot number of primary partition> -label <group name> -password <partition password >

    lunacm:> hagroup createGroup -slot 1 -label gp12 -password ******
```

4. Add the secondary members (HSM partitions). Repeat this step for all partitions to be added to the HA group.

```
lunacm:> hagroup addMember -slot <slot number of secondary partition to be added> -group <group name> -password <partition password>
<!--NeedCopy-->
```

Code:

```
lunacm:> hagroup addMember -slot 2 -group gp12 -password ******
<!--NeedCopy-->
```

5. Enable HA only mode.

```
lunacm:> hagroup HAOnly  - enable
<!--NeedCopy-->
```

6. Enable active recovery mode.

```
lunacm:.>hagroup recoveryMode  - mode active
<!--NeedCopy-->
```

7. Set auto recovery interval time (in seconds). Default is 60 seconds.

```
lunacm:.>hagroup interval  - interval <value in seconds>
<!--NeedCopy-->
```

Example:

```
lunacm:.>hagroup interval  - interval 120
<!--NeedCopy-->
```

8. Set recovery retry count. A value of -1 allows infinite number of retries.

```
lunacm:> hagroup retry -count <xxx>
<!--NeedCopy-->
```

Example:

```
lunacm:> hagroup retry -count 2
<!--NeedCopy-->
```

After configuring SafeNet HSM in HA, see Additional ADC configuration for further configuration on the ADC.

Additional ADC configuration

September 21, 2020

1) Generate a key on the HSM.
Use third party tools to create keys on the HSM.

2) Add an HSM key on the ADC.

Important!: The # character is not supported in a key name. If the key name include this character, the load key operation fails.

To add a Safenet HSM key by using the CLI:

At the command prompt, type:

```
add ssl hsmkey <KeyName> -hsmType SAFENET -serialNum <serial #> -
    password
<!--NeedCopy-->
```

where:

-keyName is the key created on the HSM by using third party tools.

-serialNum is the serial number of the partition on the HSM on which the keys are generated.

> **Note:**
>
> For HSM in a high availability setup, use the serial number of the high availability group.

-password is the password of the partition on which the keys are present.

To add a Safenet HSM key by using the GUI:

Navigate to **Traffic Management** > **SSL** > **HSM** and add an HSM key. You must specify the HSM Type as **SAFENET**.

3) Add a certificate-key pair on the ADC. You must first use a third party tool to generate a certificate associated with the key. Then, copy the certificate to the /nsconfig/ssl/ directory on the ADC.

> **Note:**
>
> The key must be an HSM key.

To add a certkey pair on the ADC by using the CLI:

At the command prompt, type:

```
add ssl certkey <CertkeyName> -cert <cert name> -hsmkey <KeyName>
<!--NeedCopy-->
```

To add a certkey pair on the ADC by using the GUI:

1. Navigate to **Traffic Management** > **SSL**.
2. In **Getting Started**, select **Install Certificate (HSM)** and create a certificate-key pair using an HSM key.

4) Create a virtual server and bind the certificate-key pair to this virtual server.

For information about creating a virtual server, click SSL virtual server configuration.

For information about adding a certificate-key pair, click Add or update a certificate-key pair.

For information about binding a certificate-key pair to an SSL virtual server, click Bind the certificate-key pair to the SSL virtual server.

NetScaler appliances in a high availability setup

September 21, 2020

You can configure a high availability (HA) setup on the NetScaler appliances with a SafeNet HSM configuration in either of the following two ways:

- First, configure a SafeNet HSM on the two nodes, using the same HSM and partition. Then create an HA pair. Finally, add the NetScaler configuration, such as keys, certificate-key pairs, and virtual servers, on the primary node.

- If a SafeNet HSM is already configured on one node with the NetScaler configuration, add a similar configuration on the other node. Copy '/var/safenet/sfgw_ident_file' from the first node to the other and restart the safenet_gw binary. After the gateway is up and running, add the nodes in an HA setup.

Limitations

September 21, 2020

1. For any changes to the HSM-related configuration in an existing setup, such as adding or removing an HSM, or creating a high availability setup, you must copy "/etc/Chrystoki.conf" to "/var/safenet/config".

2. After adding, removing, or restarting an HSM, you must restart the '/var/safenet/gateway/safenet_gw' binary. If you don't restart the gateway binary, the HSM will not serve any traffic after it is added back or after it restarts.

3. To reboot or stop the current '/var/safenet/gateway/safenet_gw' binary, use

```
kill - SIGTERM <PID>
kill - SIGINT <PID>
<!--NeedCopy-->
```

Important! Do not use `kill -9 <PID>` or `kill -6 <PID>`

4. Before removing an existing HSM from the ADC, remove, from the ADC, all the keys and certificate-key pairs that are associated with that HSM. You cannot delete these files from the ADC after you remove the HSM.

5. On a standalone NetScaler appliance, SafeNet HSMs in HA are not supported.

6. EXPORT ciphers are not supported.

7. Update certificate-key pair operation is not supported.

8. When you generate an HSM key on a third-party tool, the private and public key names must be the same. When you add the HSM key on the appliance, provide this name as the key name.

9. The # character is not supported in a key name.

10. Cluster and admin partitions are not supported.

Appendix

September 21, 2020

Sample commands with their outputs are given below.

Run the script

```
     root@ns# pwd
    /var/safenet/config
    root@ns# sh safenet_config
<!--NeedCopy-->
```

Create a certificate

```
    root@ns# cd /var/safenet/safenet/lunaclient/bin
    root@ns# ./vtl createcert -n 10.102.59.175
    Private Key created and written to: /var/safenet/safenet/lunaclient/cert/client/10.102.59.175Key.pem
    Certificate created and written to: /var/safenet/safenet/lunaclient/cert/client/10.102.59.175.pem
<!--NeedCopy-->
```

Copy the cetificate to the HSM

```
    root@ns# scp /var/safenet/safenet/lunaclient/cert/client/10.102.59.175.pem admin@10.217.2.7:
    admin@10.217.2.7's password:

    10.102.59.175.pem          100%  818     0.8KB/s   00:00
<!--NeedCopy-->
```

Copy the certificate and key from the HSM to the NetScaler appliance

```
    root@ns# scp admin@10.217.2.7:server.pem /var/safenet/safenet/lunaclient/server.2.7.pem
    admin@10.217.2.7's password:

    server.pem            100% 1164     1.1KB/s   00:01
<!--NeedCopy-->
```

Use SSH to connect to the SafeNet HSM

```
    ssh admin@10.217.2.7
    Connecting to 10.217.2.7:22...
    Connection established.
    To escape to local shell, press 'Ctrl+Alt+]'.

    Last login: Thu Jun 23 02:20:29 2016 from 10.252.243.11

    Luna SA 5.2.3-1 Command Line Shell - Copyright (c) 2001-2014 SafeNet, Inc. All rights reserved.

    [Safenet1] lunash:>hsm login


      Please enter the HSM Administrators' password:
      > *******

    'hsm login' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Register the NetScaler on the SafeNet HSM

```
    [Safenet1] lunash:>client register -client ns175 -ip 10.102.59.175

    'client register' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Assign the client a partition from the partition list

```
    [Safenet1] lunash:>client assignPartition -client ns175 -partition p2

    'client assignPartition' successful.


    Command Result : 0 (Success)
    [Safenet1] lunash:>
<!--NeedCopy-->
```

Register the HSM with its certificate on the NetScaler

```
    root@ns# ./vtl addserver -n 10.217.2.7 -c /var/safenet/safenet/lunaclient/server.2.7.pem

    New server 10.217.2.7 successfully added to server list.
<!--NeedCopy-->
```

Verify the network trust links (NTLs) connectivity between the ADC and HSM

```
    root@ns# ./vtl verify

    The following Luna SA Slots/Partitions were found:

    Slot        Serial #                  Label
    ====      ================           =====
       0              477877010          p2
<!--NeedCopy-->
```

Save the configuration

```
    root@ns# cp /etc/Chrystoki.conf /var/safenet/config/
<!--NeedCopy-->
```

Configure automatic start of the gateway daemon at boot time

```
    touch /var/safenet/safenet_is_enrolled
<!--NeedCopy-->
```

FAQ

September 21, 2020

How do I check that the SafeNet process is running?

At the NetScaler shell prompt, type:

```
ps  - aux | grep safenet_gw
<!--NeedCopy-->
```

How do I verify the network trust links (NTLs) connectivity between the ADC and HSM?

After configuring SafeNet, change directory to "/var/safenet/safenet/lunaclient/bin" and type:

```
./vtl verify
<!--NeedCopy-->
```

Troubleshooting

September 21, 2020

If the SSL feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for troubleshooting

For best results, use the following resources to troubleshoot an SSL issue on a NetScaler appliance:

- The relevant ns.log file
- The latest ns.conf file
- The messages file
- The relevant `newnslog` file
- Trace files
- A copy of the certificate files, if possible
- A copy of the key file, if possible
- The error message, if any

In addition to the preceding resources, you can use the Wireshark application customized for the NetScaler trace files to expedite troubleshooting.

Troubleshooting SSL issues

To troubleshoot an SSL issue, proceed as follows:

- Verify that the NetScaler appliance is licensed for SSL Offloading and load balancing.
- Verify that SSL Offloading and load balancing features are enabled on the appliance.
- Verify that the status of the SSL virtual server is not displayed as DOWN.
- Verify that the status of the service bound to the virtual server is not displayed as DOWN.
- Verify that a valid certificate is bound to the virtual server.
- Verify that the service is using an appropriate port, preferably port 443.

CRL refresh does not happen on the secondary node in an HA setup

The refresh does not happen because the CRL server is accessible only to the primary node through a private network.

Workaround: Add a service on the primary node with the IP address of the CRL server. This service acts as a proxy for the CRL server. When the configuration is synchronized between the nodes, CRL refresh works for both primary and secondary nodes through the service configured on the primary node.

SSL FAQs

September 21, 2020

Basic questions

- **HTTPS access to the GUI fails on a VPX instance. How do I gain access?**

A certificate-key pair is required for HTTPS access to the GUI. On a NetScaler, a certificate-key pair is automatically bound to the internal services. On an MPX or SDX appliance, the default key size is 1024 bytes, and on a VPX instance, the default key size is 512 bytes. However, most browsers today do not accept a key that is less than 1024 bytes. As a result, HTTPS access to the VPX configuration utility is blocked.

Citrix recommends that you install a certificate-key pair of at least 1024 bytes and bind it to the internal service for HTTPS access to the configuration utility or update the ns-server-certificate to 1024 bytes. You can use HTTP access to the configuration utility or the CLI to install the certificate.

- **If I add a license to an MPX appliance, the certificate-key pair binding is lost. How do I resolve this problem?**

If a license is not present on an MPX appliance when it starts, and you add a license later and restart the appliance, you might lose the certificate binding. You must reinstall the certificate and bind it to the internal service

Citrix recommends that you install an appropriate license before starting the appliance.

- **What are the various steps involved in setting up a secure channel for an SSL transaction?**

Setting up a secure channel for an SSL transaction involves the following steps:

1. The client sends an HTTPS request for a secure channel to the server.
2. After selecting the protocol and cipher, the server sends its certificate to the client.
3. The client checks the authenticity of the server certificate.
4. If any of the checks fail, the client displays the corresponding feedback.
5. If the checks pass or the client decides to continue even if a check fails, the client creates a temporary, disposable key called the *pre-master secret* and encrypts it by using the public key of the server certificate.
6. The server, upon receiving the pre-master secret, decrypts it by using the server's private key and generates the session keys. The client also generates the session keys from the pre-master secret. Thus both client and server now have a common session key, which is used for encryption and decryption of application data.

- **I understand that SSL is a CPU-intensive process. What is the CPU cost associated with the SSL process?**

The following two stages are associated with the SSL process:

- The initial handshake and secure channel setup by using the public and private key technology.
- Bulk data encryption by using the asymmetric key technology.

Both of the preceding stages can affect server performance, and they require intensive CPU processing for of the following reasons:

1. The initial handshake involves public-private key cryptography, which is very CPU intensive because of large key sizes (1024bit, 2048bit, 4096bit).
2. Encryption/decryption of data is also computationally expensive, depending on the amount of data that needs to be encrypted or decrypted.

- **What are the various entities of an SSL configuration?**

An SSL configuration has the following entities:

- Server certificate
- Certificate Authority (CA) certificate
- Cipher suite that specifies the protocols for the following tasks:

    - Initial key exchange
    - Server and client authentication
    - Bulk encryption algorithm
    - Message authentication
- Client authentication
- CRL
- SSL Certificate Key Generation Tool that enables you to create the following files:
    - Certificate request
    - Self signed certificate
    - RSA and DSA keys
    - DH parameters
- **I want to use the SSL offloading feature of the NetScaler appliance. What are the various options for receiving an SSL certificate?**

You must receive an SSL certificate before you can configure the SSL setup on the NetScaler appliance. You can use any of the following methods to receive an SSL certificate:

- Request a certificate from an authorized CA.
- Use the existing server certificate.
- Create a certificate-key pair on the NetScaler appliance.

Note: This is a test certificate signed by the test Root-CA generated by the NetScaler appliance. Test certificates signed by this Root-CA are not accepted by browsers. The browser throws a warning message stating that the server's certificate cannot be authenticated.

- For anything other than test purposes, you must provide a valid CA certificate and CA key to sign the server certificate.
- **What are the minimum requirements for an SSL setup?**

The minimum requirements for configuring an SSL setup are as follows:

- Obtain the certificates and keys.
- Create a load balancing SSL virtual server.
- Bind HTTP or SSL services to the SSL virtual server.
- Bind certificate-key pair to the SSL virtual server.
- **What are the limits for the various components of SSL?**

SSL components have the following limits:

- Bit size of SSL certificates: 4096.
- Number of SSL certificates: Depends on the available memory on the appliance.
- Maximum linked intermediate CA SSL certificates: 9 per chain.

- CRL revocations: Depends on the available memory on the appliance.
- **What are the various steps involved in the end-to-end data encryption on a NetScaler appliance?**

The steps involved in the server-side encryption process on a NetScaler appliance are as follows:

1. The client connects to the SSL VIP configured on the NetScaler appliance at the secure site.
2. After receiving the secure request, the appliance decrypts the request, applies layer 4-7 content switching techniques and load balancing policies, and selects the best available backend Web server for the request.
3. The NetScaler appliance creates an SSL session with the selected server.
4. After establishing the SSL session, the appliance encrypts the client request and sends it to the Web server by using the secure SSL session.
5. When the appliance receives the encrypted response from the server, it decrypts and re-encrypts the data, and sends the data to the client by using the client side SSL session.

The multiplexing technique of the NetScaler appliance enables the appliance to reuse SSL sessions that have been established with the Web servers. Therefore, the appliance avoids the CPU intensive key exchange, known as *full handshake*. This process reduces the overall number of SSL sessions on the server and maintains end-to-end security.

Certificates and Keys

- **Can I place the certificate and key files at any location? Is there any recommended location to store these files?**

You can store the certificate and key files on the NetScaler appliance or a local computer. However, Citrix recommends that you store the certificate and key files in the /nsconfig/ssl directory of the NetScaler appliance. The /etc directory exists in the flash memory of the NetScaler appliance. This provides portability and facilitates backup and restoration of the certificate files on the appliance.

Note: Make sure that the certificate and the key files are stored in the same directory.

- **What is the maximum size of the certificate key supported on the NetScaler appliance?**

A NetScaler appliance running a software release earlier than release 9.0 supports a maximum certificate key size of 2048 bits. Release 9.0 and later support a maximum certificate key size of 4096 bits. This limit is applicable to both RSA and DSA certificates.

An MPX appliance supports certificates from 512-bits up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service

- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back end server
- 4096-bit client certificate (if client authentication is enabled on the virtual server)

A virtual appliance supports certificates from 512-bits up to the following sizes:

- 4096-bit server certificate on the virtual server
- 4096-bit client certificate on the service
- 4096-bit CA certificate (includes intermediate and root certificates)
- 4096-bit certificate on the back end server from release 12.0-56.x. Older releases support 2048-bit certificates.
- 2048-bit client certificate (if client authentication is enabled on the virtual server) from release 12.0-56.x.
- **What is the maximum size of the DH parameter supported on the NetScaler appliance?**

The NetScaler appliance supports a DH parameter of maximum 2048 bits.

- **What is the maximum certificate-chain length, that is, the maximum number of certificates in a chain, supported on a NetScaler appliance?**

A NetScaler appliance can send a maximum of 10 certificates in a chain when sending a server certificate message. A chain of the maximum length includes the server certificate and nine intermediate CA certificates.

- **What are the various certificate and key formats supported on the NetScaler appliance?**

The NetScaler appliance supports the following certificate and key formats:

- Privacy Enhanced Mail (PEM)
- Distinguished Encoding Rule (DER)
- **Is there a limit for the number of certificates and keys that I can install on the NetScaler appliance?**

No. The number of certificates and keys that can be installed is limited only by the available memory on the NetScaler appliance.

- **I have saved the certificate and key files on the local computer. I want to transfer these files to the NetScaler appliance by using the FTP protocol. Is there any preferred mode for transfering these files to the NetScaler appliance?**

Yes. If using the FTP protocol, you should use binary mode to transfer the certificate and key files to the NetScaler appliance.

Note: By default, FTP is disabled. Citrix recommends using the SCP protocol for transferring certificate and key files. The configuration utility implicitly uses SCP to connect to the appliance.

- **What is the default directory path for the certificate and key?**

The default directory path for the certificate and key is '/nsconfig/ssl'.

- **When adding a certificate and key pair, what happens if I do not specify an absolute path to the certificate and key files?**

When adding a certificate and key pair, if you do not specify an absolute path to the certificate and key files, the NetScaler appliance searches the default directory, /nsconfig/ssl, for the specified files and attempts to load them to the kernel. For example, if the cert1024.pem and rsa1024.pem files are available in the /nsconfig/ssl directory of the appliance, both of the following commands are successful:

```
add ssl certKey cert1 -cert cert1204.pem -key rsa1024.pem
<!--NeedCopy-->
```

```
add ssl certKey cert1 -cert /nsconfig/ssl/cert1204.pem -key /nsconfig/ssl/rsa1024.pem
<!--NeedCopy-->
```

- **I have configured a high availability setup. I want to implement the SSL feature on the setup. How should I handle the certificate and key files in a high availability setup?**

In a high availability setup, you must store the certificate and key files on both the primary and the secondary NetScaler appliance. The directory path for the certificate and key files must be the same on both appliances before you add an SSL certificate-key pair on the primary appliance.

Thales nShield® HSM

- **When integrating with Thales nShield® HSM, do we have to keep in mind any specific configuration when adding the NetScaler appliance to HA?**

You must configure the same Thales device(s) on both the nodes in HA. Thales configuration commands don't synchronize in HA. For information about the prerequisites for Thales nShield® HSM, see Prerequisites.

- **Do we have to individually integrate both the appliances with Thales nShield® HSM and RFS? Do we need to do this before or after the HA setup?**

You can complete the integration before or after the HA setup. However, if the integration is done after the HA setup, the keys imported on the primary node prior to configuring the secondary node are not synced to the secondary node. Therefore, Citrix recommends Thales integration before the HA setup.

- **Do we have to import the key into both the primary and secondary NetScaler appliances, or are the keys synchronized from the primary node to the secondary node?**

If Thales is integrated on both devices before forming the HA, the keys are automatically synchronized from RFS in the process of integration.

- **Given that the HSM is not on the NetScaler appliance, but on Thales, what happens to the keys and certificates when a node fails and is replaced?**

If a node fails, it is possible to synchronize the keys and certificates to the new node, by first integrating Thales on the new node and then running the following commands:

```
sync ha files ssl
force ha sync
<!--NeedCopy-->
```

The certificates are synchronized and added if the keys are synchronized in the process of integrating Thales.

Ciphers

- **What is a NULL-Cipher?**

Ciphers with no encryption are known as NULL-Ciphers. For example, NULL-MD5 is a NULL-Cipher.

- **Are the NULL-Ciphers enabled by default for an SSL VIP or an SSL service?**

No. NULL-Ciphers are not enabled by default for an SSL VIP or an SSL service.

- **What is the procedure to remove NULL-Ciphers?**

To remove the NULL-Ciphers from an SSL VIP, run the following command:

```
bind ssl cipher <SSL_VIP> REM NULL
<!--NeedCopy-->
```

To remove the NULL-Ciphers from an SSL Service, run the following command:

```
bind ssl cipher <SSL_Service> REM NULL -service
<!--NeedCopy-->
```

- **What are the various cipher aliases supported on the NetScaler appliance?**

To list the cipher aliases supported on the appliance, at the command prompt, type:

```
sh cipher
<!--NeedCopy-->
```

- **What is the command to display all the predefined ciphers of the NetScaler appliance?**

To display all the predefined ciphers of the NetScaler appliance, at the CLI, type:

```
show ssl cipher
<!--NeedCopy-->
```

- **What is the command to display the details of an individual cipher of the NetScaler appliance?**

To display the details of an individual cipher of the NetScaler appliance, at the CLI, type:

```
show ssl cipher <Cipher_Name/Cipher_Alias_Name/Cipher_Group_Name>
<!--NeedCopy-->
```

Example:

```
show cipher SSL3-RC4-SHA
     1) Cipher Name: SSL3-RC4-SHA
     Description: SSLv3 Kx=RSA Au=RSA Enc=RC4(128)
    Mac=SHA1
     Done
<!--NeedCopy-->
```

- **What is the significance of adding the predefined ciphers of the NetScaler appliance?**

Adding the predefined ciphers of the NetScaler appliance causes the NULL-Ciphers to get added to an SSL VIP or an SSL service.

Certificates

- **Is the distinguished name in a client certificate available for the length of the user session?**

Yes. You can access the distinguished name of the client certificate in subsequent requests during the length of the user session, that is even after the SSL handshake is complete and the certificate is not sent again by the browser. To do this, use a variable and an assignment as detailed in the following sample configuration:

Example:

```
add ns variable v2 -type "text(100)"

add ns assignment a1 -variable "$v2" -set       "CLIENT.SSL.CLIENT_CERT
   .SUBJECT.TYPECAST_NVLIST_T('=','/').VALUE("CN")"

add rewrite action act1 insert_http_header subject "$v2"  // example:
   to insert the distinguished name in the header

add rewrite policy pol1 true a1

add rewrite policy pol2 true act1

bind rewrite global pol1 1 next -type RES_DEFAULT

bind rewrite global pol2 2 next -type RES_DEFAULT

set rewrite param -undefAction RESET
<!--NeedCopy-->
```

- **Why do I need to bind the server certificate?**

Binding the server certificates is the basic requirement for enabling the SSL configuration to process SSL transactions.

To bind the server certificate to an SSL VIP, at the CLI, type:

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

To bind the server certificate to an SSL service, at the CLI, type:

```
bind ssl service <serviceName> -certkeyName <cert_name>
<!--NeedCopy-->
```

- **How many certificates can I bind to an SSL VIP or an SSL service?**

On a NetScaler virtual appliance, you can bind a maximum of three certificates to an SSL VIP or an SSL service, one each of type RSA,ECDSA, and DSA. On a NetScaler MPX (N2) or MPX-FIPS appliance, if SNI is enabled, you can bind multiple server certificates of type RSA. If SNI is disabled, you can bind a maximum of one certificate of type RSA. On a NetScaler MPX (N3) appliance, you can bind a maximum of two certificates to an SSL VIP or an SSL service, one each of type RSA and type ECDSA.

Note: DSA certificates are not supported on MPX or MPX-FIPS platforms.

- **Does SNI support Subject Alternative Name (SAN) certificates?**

No. On a NetScaler appliance, SNI is not supported with a SAN extension certificate.

- **What happens if I unbind or overwrite a server certificate?**

When you unbind or overwrite a server certificate, all the connections and SSL sessions created by using the existing certificate are terminated. When you overwrite an existing certificate, the following message appears:

```
ERROR:

Warning: Current certificate replaces the previous binding.
```

```
<!--NeedCopy-->
```

- **How do I install an intermediate certificate on NetScaler and link to a server certificate?**

See the article at
http://support.citrix.com/article/ctx114146 for information about installing an intermediate certificate.

- **Why am I am getting a "resource already exists" error when I try to install a certificate on the NetScaler?**

See the article at
http://support.citrix.com/article/CTX117284 for instructions for resolving the "resource already exists" error.

- **I want to create a server certificate on a NetScaler appliance to test and evaluate the product. What is the procedure to create a server certificate?**

Perform the following procedure to create a test certificate.

Note: A certificate created with this procedure cannot be used to authenticate all the users and browsers. After using the certificate for testing, you should obtain a server certificate signed by an authorized Root CA.

To create a self-signed server certificate:

1. To create a Root CA certificate, at the CLI, type:

   ```
   create ssl rsakey /nsconfig/ssl/test-ca.key 1024

   create ssl certreq /nsconfig/ssl/test-ca.csr -keyfile /nsconfig/
       ssl/test-ca.key

   Enter the required information when prompted, and then type the
       following command:

   create ssl cert /nsconfig/ssl/test-ca.cer /nsconfig/ssl/test-ca.
       csr ROOT_CERT -keyfile /nsconfig/ssl/test-ca.key
   <!--NeedCopy-->
   ```

2. Perform the following procedure to create a server certificate and sign it with the root CA certificate that you just created

   a) To create the request and the key, at the CLI, type:

   ```
   create ssl rsakey /nsconfig/ssl/test-server.key 1024

   ```

```
    create ssl certreq /nsconfig/ssl/test-server.csr -keyfile /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

b) Enter the required information when prompted.

c) To create a serial-number file, at the CLI, type:

```
shell
 # echo '01' >
/nsconfig/ssl/serial.txt
 # exit
<!--NeedCopy-->
```

d) To create a server certificate signed by the root CA certificate created in step 1, at the CLI, type:

```
create ssl cert /nsconfig/ssl/test-server.cer /nsconfig/ssl/test-server.csr SRVR_CERT -CAcert /nsconfig/ssl/test-ca.cer -CAkey /nsconfig/ssl/test-ca.key -CAserial /nsconfig/ssl/serial.txt
<!--NeedCopy-->
```

e) To create a NetScaler certkey, which is the in-memory object that holds the server certificate information for SSL handshakes and bulk encryption, at the CLI, type:

```
add ssl certkey test-certkey -cert /nsconfig/ssl/test-server.cer -key /nsconfig/ssl/test-server.key
<!--NeedCopy-->
```

f) To bind the certkey object to the SSL virtual server, at the CLI, type:

```
bind ssl vserver <vServerName> -certkeyName <cert_name>
<!--NeedCopy-->
```

- **I have received a NetScaler appliance on which NetScaler software release 9.0 is installed. I have noticed an additional license file on the appliance. Is there any change in the licensing policy starting with Citrix NetScaler software release 9.0?**

Yes. Starting with Citrix NetScaler software release 9.0, the appliance might not have a single license file. The number of license files depends on the NetScaler software release edition. For example, if you have installed the Enterprise edition, you might need additional license files for the full functionality of the various features. However, if you have installed the Platinum edition, the appliance has only one license file.

- **How do I export the certificate from Internet Information Service (IIS)?**

There are many ways to do this, but by using the following method the appropriate certificate and private key for the Web site are exported. This procedure must be performed on the actual IIS server.

1. Open the Internet Information Services (IIS) Manager administration tool.
2. Expand the Web Sites node and locate the SSL-enabled Web site that you want to serve through the NetScaler appliance.
3. Right-click this Web site and click Properties.
4. Click the Directory Security tab and, in the Secure Communications section of the window, select the View Certificate box.
5. Click the Details tab, and then click Copy to File.
6. On the Welcome to the Certificate Export Wizard page, click Next.
7. Select Yes, export the private key and click Next.

   Note: The private key MUST be exported for SSL Offload to work on the NetScaler.

8. Make sure that the Personal Information Exchange -PKCS #12 radio button is selected, and select *only* the Include all certificates in the certification path if possible check box. Click Next.
9. Enter a password and click Next.
10. Enter a file name and location, and then click Next. Give the file an extension of .PFX.
11. Click Finish.

- **How do I convert the PKCS#12 certificate and install it on the NetScaler?**

1. Move the exported .PFX certificate file to a location from where it may be copied to the NetScaler (that is, to a machine that permits SSH access to the management interface of a NetScaler appliance). Copy the certificate to the appliance by using a secure copy utility such as SCP.
2. Access the BSD shell and convert the certificate (for example, cert.PFX) to .PEM format:

   ```
   root@ns# openssl pkcs12 -in cert.PFX -out cert.PEM
   <!--NeedCopy-->
   ```

3. To make sure that the converted certificate is in correct x509 format, verify that the following command produces no error:

   ```
   root@ns# openssl x509 -in cert.PEM -text
   <!--NeedCopy-->
   ```

4. Verify that the certificate file contains a private key. Begin by issuing the following command:

   ```
   root@ns# cat cert.PEM

   ```

```
Verify that the output file includes an RSA PRIVATE KEY section.

-----BEGIN RSA PRIVATE KEY-----
Mkm^s9KMs9023pz/s...
-----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

The following is another example of an RSA PRIVATE KEY section:

```
    Bag Attributes
    1.3.6.1.4.1.311.17.2: <No Values>
    localKeyID: 01 00 00 00
    Microsoft CSP Name: Microsoft RSA SChannel Cryptographic
    Provider
    friendlyName:
    4b9cef4cc8c9b849ff5c662fd3e0ef7e_76267e3e-6183-4d45-886e-6e067297b38f

    Key Attributes
    X509v3 Key Usage: 10
    -----BEGIN RSA PRIVATE KEY-----
    Proc-Type: 4,ENCRYPTED
    DEK-Info: DES-EDE3-CBC,43E7ACA5F4423968
    pZJ2SfsSVqMbRRf6ug37Clua5gY0Wld4frPIxFXyJquUHr31dilW5ta3hbIaQ+Rg

    ... (more random characters)
    v8dMugeRplkaH2Uwt/mWBk4t71Yv7GeHmcmjafK8H8iW80ooPO3D/ENV8X4U/tlh

    5eU6ky3WYZ1BTy6thxxLlwAullynVXZEflNLxq1oX+ZYl6djgjE3qg==
    -----END RSA PRIVATE KEY-----
<!--NeedCopy-->
```

The following is a SERVER CERTIFICATE section:

```
    Bag Attributes
    localKeyID: 01 00 00 00
    friendlyName: AG Certificate
    subject=/C=AU/ST=NSW/L=Wanniassa/O=Dave Mother
    Asiapacific/OU=Support/CN=davemother.food.lan
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIFiTCCBHGgAwIBAgIKCGryDgAAAAAHzANBgkqhkiG9w0BAQUFADA8MRMwEQYK

    ... (more random characters) 5
       pLDWYVHhLkA1pSxvFjNJHRSIydWHc5ltGyKqIUcBezVaXyel94pNSUYx07NpPV
       /

    MY2ovQyQZM8gGe3+lGFum0VHbv/y/gB9HhFesog=
    -----END CERTIFICATE-----
<!--NeedCopy-->
```

The following is an INTERMEDIATE CA CERTIFICATE section:

```
    Bag Attributes: <Empty Attributes>
    subject=/DC=lan/DC=food/CN=hotdog
    issuer=/DC=lan/DC=food/CN=hotdog
    -----BEGIN CERTIFICATE-----
    MIIESDCCAzCgAwIBAgIQah20fCRYTY9LRXYMIRaKGjANBgkqhkiG9w0BAQUFADA8


    ... (more random characters)
       Nt0nksawDnbKo86rQcNnY5xUs7c7pj2zxj/IOsgNHUp5W6dDI9pQoqFFaDk
       =

    -----END CERTIFICATE-----
<!--NeedCopy-->
```

Further Intermediate CA certificates may follow, depending on the certification path of the exported certificate.

5. Open the .PEM file in a text editor
6. Locate the first line of the .PEM file and the first instance of the following line, and copy those two lines and all the lines between them:

```
        -----END CERTIFICATE-----

        Note: Make sure that last copied line is the first
        -----END CERTIFICATE----- line in the .PEM file.
<!--NeedCopy-->
```

1. Paste the copied lines into a new file. Call the new file something intuitive, such as cert-key.pem. This is the certificate-key pair for the server hosting the HTTPS service. This file should contain both the section labeled RSA PRIVATE KEY and the section labeled SERVER CERTIFICATE in the example above.

   Note: The certificate-key pair file contains the private key and must therefore be kept secure.

2. Locate any subsequent sections beginning with —–BEGIN CERTIFICATE—– and ending with —END CERTIFICATE—–, and copy each such section to a separate new file.

   These sections correspond to certificates of trusted CAs that have been included in the certification path. These sections should be copied and pasted into new individual files for these certificates. For example, the INTERMEDIATE CA CERTIFICATE section of the example above should be copied and pasted into a new file).

   For multiple intermediate CA certificates in the original file, create new files for each intermediate CA certificate in the order in which they appear in the file. Keep track (using appropriate filenames) of the order in which the certificates appear, as they need to be linked together in the correct order in a later step.

3. Copy the certificate-key file (cert-key.pem) and any additional CA certificate files into the /nsconfig/ssl directory on the NetScaler appliance.
4. Exit the BSD shell and access the NetScaler prompt.
5. Follow the steps in "Install the certificate-key files on the appliance" to install the key/certificate once uploaded on the device.

- **How do I convert the PKCS#7 certificate and install it on the NetScaler appliance?**

You can use OpenSSL to convert a PKCS #7 Certificate to a format recognizable by the NetScaler appliance. The procedure is identical to the procedure for PKCS #12 certificates, except that you invoke OpenSSL with different parameters. The steps for converting PKCS #7 certificates are as follows:

1. Copy the certificate to the appliance by using a secure copy utility, such as SCP.
2. Convert the certificate (for example, cert.P7B) to PEM format:

   ```
   openssl pkcs7 -inform DER -in cert.p7b -print_certs -text -out
       cert.pem
   <!--NeedCopy-->
   ```

3. Follow steps 3 through 7 as described in the answer for PKCS #12 certificates.
   Note: Before loading the converted PKCS #7 certificate to the appliance, be sure to verify that it contains a private key, exactly as described in step 3 for the PKCS #12 procedure. PKCS #7 certificates, particularly those exported from IIS, do not typically contain a private key.

- **When I bind a cipher to a virtual server or service by using the bind cipher command, I see the error message "Command deprecated."**

The command for binding a cipher to a virtual server or service has changed.

Use the bind ssl vserver <vsername> -ciphername <ciphername> command to bind an SSL cipher to an SSL virtual server.

Use the bind ssl service <serviceName> -ciphername <ciphername> command to bind an SSL cipher to an SSL service.

Note: New ciphers and cipher groups are added to the existing list and not replaced.

- **Why can't I create a new cipher group and bind ciphers to it by using the add cipher command?**

The add cipher command functionality has changed in release 10. The command only creates a cipher group. To add ciphers to the group, use the bind cipher command.

OpenSSL

- **How do I use OpenSSL to convert certificates between PEM and DER?**

To use OpenSSL, you must have a working installation of the OpenSSL software and be able to execute Openssl from the command line.

x509 certificates and RSA keys can be stored in a number of different formats.

Two common formats are DER (a binary format used primarily by Java and Macintosh platforms) and PEM (a base64 representation of DER with header and footer information, which is used primarily by UNIX and Linux platforms). There is also an obsolete NET (Netscape server) format that was used by earlier versions of IIS (up to and including 4.0) and various other less common formats that are not covered in this article.

A key and the corresponding certificate, as well as the root and any intermediate certificates, can also be stored in a single PKCS#12 (.P12, .PFX) file.

Procedure

Use the Openssl command to convert between formats as follows:

1. To convert a certificate from PEM to DER:

   ```
   x509 -in input.crt -inform PEM -out output.crt -outform DER
   <!--NeedCopy-->
   ```

2. To convert a certificate from DER to PEM:

   ```
   x509 -in input.crt -inform DER -out output.crt -outform PEM
   <!--NeedCopy-->
   ```

3. To convert a key from PEM to DER:

   ```
   rsa -in input.key -inform PEM -out output.key -outform DER
   <!--NeedCopy-->
   ```

4. To convert a key from DER to PEM:

   ```
   rsa -in input.key -inform DER -out output.key -outform PEM
   <!--NeedCopy-->
   ```

   Note: If the key you are importing is encrypted with a supported symmetric cipher, you are prompted to enter the pass-phrase.

   Note: To convert a key to or from the obsolete NET (Netscape server) format, substitute NET for PEM or DER as appropriate. The stored key is encrypted in a weak unsalted RC4 symmetric cipher, so a pass-phrase will be requested. A blank pass-phrase is acceptable.

System Limits

- **What are the important numbers to remember?**

1. Create Certificate Request:
   - Request File Name: Maximum 63 characters
   - Key File Name: Maximum 63 characters
   - PEM Passphrase (For Encrypted Key): Maximum 31 characters
   - Common Name: Maximum 63 characters
   - City: Maximum 127 characters
   - Organization Name: Maximum 63 characters
   - State/Province Name: Maximum 63 characters
   - Email Address: Maximum 39 Characters
   - Organization Unit: Maximum 63 characters
   - Challenge Password: Maximum 20 characters
   - Company Name: Maximum 127 characters
2. Create Certificate:
   - Certificate File Name: Maximum 63 characters
   - Certificate Request File Name: Maximum 63 characters
   - Key File Name: Maximum 63 characters
   - PEM Passphrase: Maximum 31 characters
   - Validity Period: Maximum 3650 days
   - CA Certificate File Name: Maximum 63 characters
   - CA Key File Name: Maximum 63 characters
   - PEM Passphrase: Maximum 31 characters
   - CA Serial Number File: Maximum 63 characters
3. Create and Install a Server Test Certificate:
   - Certificate File Name: Maximum 31 characters

- Fully Qualified Domain Name: Maximum 63 characters

4. Create Diffie-Hellman (DH) key:
    - DH Filename (with path): Maximum 63 characters
    - DH Parameter Size: Maximum 2048 bits
5. Import PKCS12 key:
    - Output File Name: Maximum 63 characters
    - PKCS12 File Name: Maximum 63 characters
    - Import Password: Maximum 31 characters
    - PEM Passphrase: Maximum 31 characters
    - Verify PEM Passphrase: Maximum 31 characters
6. Export PKCS12
    - PKCS12 File Name: Maximum 63 characters
    - Certificate File Name: Maximum 63 characters
    - Key File Name: Maximum 63 characters
    - Export Password: Maximum 31 characters
    - PEM Passphrase: Maximum 31 characters
7. CRL Management:
    - CA Certificate File Name: Maximum 63 characters
    - CA Key File Name: Maximum 63 characters
    - CA Key File Password: Maximum 31 characters
    - Index File Name: Maximum 63 characters
    - Certificate File Name: Maximum 63 characters
8. Create RSA Key:
    - Key Filename: Maximum 63 characters
    - Key Size: Maximum 4096 bits
    - PEM Passphrase: Maximum 31 characters
    - Verify Passphrase: Maximum 31 characters
9. Create DSA Key:
    - Key Filename: Maximum 63 characters
    - Key Size: Maximum 4096 bits
    - PEM Passphrase: Maximum 31 characters
    - Verify Passphrase: Maximum 31 characters
10. Change advanced SSL settings:
    - Maximum CRL memory size: Maximum 1024 Mbytes
    - Encryption trigger timeout (10 mS ticks): Maximum 200

- Encryption trigger packet count: Maximum 50
- OCSP cache size: Maximum 512 Mbytes

11. Install Certificate:
    - Certificate-Key pair Name: Maximum 31 characters
    - Certificate File Name: Maximum 63 characters
    - Private Key File Name: Maximum 63 characters
    - Password: Maximum 31 characters
    - Notification Period: Maximum 100
12. Create Cipher Group:
    - Cipher Group Name: Maximum 39 characters
13. Create CRL:
    - CRL Name: Maximum 31 characters
    - CRL File: Maximum 63 characters
    - URL: Maximum 127 characters
    - Base DN: Maximum 127 characters
    - Bind DN: Maximum 127 characters
    - Password: Maximum 31 characters
    - Day(s): Maximum 31
14. Create SSL Policy:
    - Name: Maximum 127 characters
15. Create SSL Action:
    - Name: Maximum 127 characters
16. Create OCSP Responder:
    - Name: Maximum 32 characters
    - URL: Maximum 128 characters
    - Batching Depth: Maximum 8
    - Batching Delay: Maximum 10000
    - Produced At Time Skew: Maximum 86400
    - Request Time-out: Maximum120000
17. Create Virtual Server:
    - Name: Maximum 127 characters
    - Redirect URL: Maximum 127 characters
    - Client Time-out: Maximum 31536000 secs
18. Create Service:

  - Name: Maximum 127 characters
  - Idle Time-out (secs):
    Client: Maximum 31536000
    Server: Maximum 31536000

19. Create Service Group:

    - Service Group Name: Maximum 127 characters
    - Server ID: Maximum 4294967295
    - Idle Time-out (secs):
      Client: Maximum value 31536000
      Server: Maximum 31536000

20. Create Monitor:

    - Name: Maximum 31 characters

21. Create Server:

    - Server Name: Maximum 127 characters
    - Domain Name: Maximum 255 characters
    - Resolve Retry: Maximum 20939 secs

Security

September 21, 2020

The following topics cover configuration and installation information for NetScaler security features. Most of these features are policy based.

Authentication, Authorization, Auditing (AAA)	Keeps unauthorized users out of the network, denies users access to tasks for which they are not authorized, and tracks the resources used during user sessions.
Application Firewall	Prevents security breaches, data loss, and possible unauthorized modifications to web sites that access sensitive business or customer information.
Content Filtering	Blocks inappropriate HTML requests, preventing the requests from reaching the Web servers.

HTTP Denial-of-Service Protection	Prevents hackers from attacking your Web site with large numbers of HTTP requests.
Priority Queuing	Detects high-priority connections and allows those connections to proceed ahead of other connections, guaranteeing unimpeded access to those users.
SureConnect	Services all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.
Surge Protection	Detects any rapid rise in connection attempts and adjusts the rate at which connections are allowed to proceed to the server, preventing server overload.

Content filtering

September 21, 2020

Content filtering can do some of the same tasks as the Application Firewall, and is a less CPU-intensive tool. It is limited, however, to examining the header portion of the HTTP request or response and to performing a few simple actions on connections that match. If you have a complex Web site that makes extensive use of scripts and accesses back-end databases, the Application Firewall may be the better tool for protecting that Web site. For more information about the Application Firewall, see Application Firewall.

Content filtering is based on regular expressions that you can apply to either HTTP requests or HTTP responses. To block requests from a particular site, for example, you could use an expression that compares each request's URL to the URL specified in the expression. The expression is part of a policy, which also specifies an action to be performed on requests or responses that match the expression. For example, an action might drop a request or reset the connection.

Following are some examples of things you can do with content filtering policies:

- Prevent users from accessing certain parts of your Web sites unless they are connecting from authorized locations.
- Prevent inappropriate HTTP headers from being sent to your Web server, possibly breaching

security.
- Redirect specified requests to a different server or service.

To configure content filtering, once you have made sure that the feature is enabled, you configure filtering actions for your servers to perform on selected connections (unless the predefined actions are adequate for your purposes). Then you can configure policies to apply the actions to selected connections. Your policies can use predefined expressions, or you can create your own. To activate the policies you configured, you bind them either globally or to specific virtual servers.

Enabling content filtering

September 21, 2020

By default, content filtering is enabled on NetScaler appliances running the NetScaler operating system 8.0 or above. If you are upgrading an existing appliance from an operating system version earlier than 8.0, you must update the licenses before you can use content filtering, and you may need to enable the content filtering feature itself manually.

Enable content filtering by using the CLI

At the command prompt, type the following commands to enable content filtering and verify the configuration:

```
enable ns feature ContentFiltering
show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature ContentFiltering
 Done
show ns feature

        Feature                         Acronym             Status
        -------                         -------             ------
 1)     Web Logging                     WL                  ON
 2)     Surge Protection                SP                  OFF
 .
 .
 .
 .
 11)    Http DoS Protection             HDOSP               OFF
 12)    Content Filtering              CF                   ON
 .
 .
 23)    HTML Injection                 HTMLInjection        ON
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

Enable content by filtering by using the GUI

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Configure basic features.
3. In the Configure Basic Features pane, select the Content Filter check box, and then click OK.

Configure a content filtering action

September 21, 2020

After you enable the content filtering feature, you create one or more actions to tell your NetScaler appliance how to handle the connections it receives.

Content filtering supports the following actions for HTTP requests:

- **Add:** Adds the specified HTTP header before sending the request to the web server.
- **Reset:** Terminates the connection, sending the appropriate termination notice to the user's browser.
- **Forward:** Redirects the request to the designated service.
- **Drop:** Silently deletes the request, without sending a response to the user's browser.
- **Corrupt:** Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the request to the server.

Content filtering supports the following actions for HTTP responses:

- **Add:** Adds the specified HTTP header before sending the response to the user's browser.
- **ErrorCode:** Returns the designated HTTP error code to the user's browser.
- **Corrupt:** Modifies the designated HTTP header in a manner that prevents it from performing the function it was intended to perform, then sends the response to the user's browser.

Configure a content filtering action by using the CLI

At the command prompt, type the following commands to configure a Content Filtering action and verify the configuration:

```
add filter action <name> <qualifier> [<serviceName>] [<value>] [<
    respCode>] [<page>]
show filter action <name>
<!--NeedCopy-->
```

Example:

```
add filter action act_drop Drop
 Done
show filter action act_drop
1)      Name: act_drop   Filter Type: drop
 Done
<!--NeedCopy-->
```

Configure a content filtering action by using the GUI

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, do one of the following:
   - To create a new action, click Add.
   - To modify an existing action, select the action, and then click Open.
3. In the Add Filter Action or Configure Filter Action dialog box, specify values for the parameters:
   - Action Name*—name
   - Qualifier*—qualifier (Determines which of the following parameters you can configure)
   - Service Name—servicename
   - HeaderName:Value—value
   - Response Code—respcode
   - Response Page—page
4. Fill in any other required information. For example, if you are configuring an action to send an HTTP error code, you must choose the appropriate error code from a drop-down list. If necessary, you can then modify the text of the error message, which is displayed beneath the drop-down list.
5. Click Create or OK, and then click Close. The Actions list displays the action you configured, and a message in the status bar indicates that your action has been created.

Configure a content filtering policy

September 21, 2020

To implement content filtering, you must configure at least one policy to tell your NetScaler appliance how to distinguish the connections you want to filter. You must first have configured at least one filtering action, because when you configure a policy, you associate it with an action.

Content filtering policies examine a combination of one or more of the following elements to select requests or responses for filtering:

- **URL:** The URL in the HTTP request.
- **URL query:** Only the query portion of the URL, which is the portion after the query (?) symbol.
- **URL token:** Only the tokens in the URL, if any, which are the parts that begin with an ampersand (&) and consist of the token name, followed by an equals sign (=), followed by the token value.
- **HTTP method:** The HTTP method used in the request, which is usually GET or POST, but can be any of the eight defined HTTP methods.
- **HTTP version:** The HTTP version in the request, which is usually HTTP 1.1.
- **Standard HTTP header:** Any of the standard HTTP headers defined in the HTTP 1.1 specification.
- **Standard HTTP header value:** The value portion of the HTTP header, which is the portion after the colon and space (:).
- **Custom HTTP header:** A non-standard HTTP header issued by your Web site or that appears in a user request.
- **Custom header value:** The value portion of the custom HTTP header, which (as with the standard HTTP header) is the portion after the colon and space (:).
- **Client Source IP:** The IP from which the client request was sent.

Content filtering policies use the simpler of two NetScaler expressions languages, called classic expressions. For a complete description of classic expressions, how they work, and how to configure them manually, see "Policies and Expressions."

Note: Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

Configure a content filtering policy by using the CLI

At the command prompt, type the following commands to configure a content filtering policy and verify the configuration:

```
add filter policy \<name\> -rule \<expression\> (-reqAction \<action\>
    | -resAction \<string\>
show filter policy \<name\>
<!--NeedCopy-->
```

Example:

```
add filter policy cf-pol -rule "REQ.HTTP.URL CONTAINS http://abc.com" -
    reqaction DROP
 Done
show filter policy cf-pol
1)       Name: cf-pol    Rule: REQ.HTTP.URL CONTAINS http://abc.com
         Request action: DROP
         Response action:
         Hits: 0
 Done
<!--NeedCopy-->
```

Configure a content filtering policy by using the GUI

1. Navigate to Security > Protection Features > Filter.
2. Navigate to Protection Features > Filter.
3. In the details pane, to create a new policy, click Add.
4. If you are creating a new policy, in the Create Filter Policy dialog box, in the Filter Name text box, type a name for your new policy.
5. Select either Request Action or Response Action to activate the drop-down list to the right of that item.
6. Click the down arrow to the right of the drop-down list and select the action to be performed on the request or response. The default choices are RESET and DROP. Any other actions you have created will also appear in this list.
   Note: You can also click
   New to create a new Content Filtering action, or
   Modify to modify an existing Content Filtering action. You can only modify actions you created; the default actions are read-only.
7. If you want to use a predefined expression (or named expression) to define your policy, choose one from the Named Expressions list.
   a) Click the down arrow to the right of the first Named Expressions drop-down list, and choose the category of named expressions that contains the named expression you want to use.
   b) Click the down arrow to the right of the second Named Expressions drop-down list, and choose the named expression you want. As you choose a named expression, the regular

    expression definition of that named expression appears in the Preview Expression pane beneath the Named Expression list boxes.

    c) Click Add Expression to add that named expression to the Expression list.
    Note: You should perform either this step or step 7, but not both.

8. If you want to create a new expression to define your policy, use the Expression Editor.
    a) Click the Add button. The Add Expression dialog box appears.
    b) In the Add Expression dialog box, choose the type of connection you want to filter. The Flow Type is set to REQ by default, which tells the NetScaler appliance to look at incoming connections, or requests. If you want to filter outgoing connections (responses), you click the right arrow beside the drop-down list and choose RES.
    c) If the Protocol is not already set to HTTP, click the down arrow to the right of the Protocol drop-down list and choose HTTP.
    Note: In the NetScaler classic expressions language, "HTTP" includes HTTPS requests, as well.
    d) Click the down arrow to the right of the Qualifier drop-down list, and then choose a qualifier for your expression. Your choices are:
        - **METHOD:** The HTTP method used in the request.
        - **URL:** The contents of the URL header.
        - **URLTOKENS:** The URL tokens in the HTTP header.
        - **VERSION:** The HTTP version of the connection.
        - **HEADER:** The header portion of the HTTP request.
        - **URLLEN:** The length of the contents of the URL header.
        - **URLQUERY:** The query portion of the contents of the URL header.
        - **URLQUERYLEN:** The length of the query portion of the URL header.
        The contents of the remaining list boxes change to the choices appropriate to the Qualifier you pick. For example, if you choose HEADER, a text field labeled Header Name* appears below the Flow Type list box.
    e) Click the down arrow to the right of the Operator drop-down list, and choose an operator for your expression. Your choices will vary depending on the Protocol you chose in the preceding step. The following list includes all of the operators:
        - **==:** Matches the following text string exactly.
        - **!=:** Does not exactly match the following text string.
        - **>:** Is greater than the following integer.
        - **CONTAINS:** Contains the following text string.
        - **CONTENTS:** The contents of the designated header, URL, or URL query.
        - **EXISTS:** The specified header or query exists.
        - **NOTCONTAINS:** Does not contain the following text string.
        - **NOTEXISTS:** The specified header or query does not exist.
    f) If the Value text box is visible, type the appropriate string or number. If you are testing a

    string in any way, type the string into the Value text box. If you are testing an integer in any way, type the integer into the Value text box.
    g) If you chose HEADER as the Protocol, type the header you want in the Header Name* text box.
    h) Click OK to add your expression to the Expressions list.
    i) Repeat steps B through H to create any additional expressions you want for your profile.
    j) Click Close to close the Expressions Editor.

9. If you created a new expression, in the Expression frame select an option from the Match Any Expression drop-down list. Your choices are:
    - Match Any Expression. If a request matches any expression in the Expressions list, the request matches this policy.
    - Match All Expressions If a request matches all expressions in the Expressions list, the request matches this policy. If it does not matchall of them, it does not match this policy.
    - Tabular Expression Switches the Expressions list to a tabular format with three columns. In the first column you can place a BEGIN [(] operator. The second column contains the expressions you have selected or created. In the third column, you can place any of the other operators in the following list, to create complex policy groups in which each group can be configured for match any expression or match all expressions.
    - The AND [&&] operator tells the appliance to require that are quest match both the current expression and the following expression.

The OR [	] operator tells the appliance to require that arequest match either the current expression or the following expression, or both. Only if the request does not match either expression does it not match the policy.

    -
    - The END [)] operator tells the appliance that this is the last expression in this expression group or policy.
      Note: The Tabular format allows you to create a complex policy that contains both "Match Any Expression" and "Match All Expressions" on a per-expression basis. You are not limited to just one or the other.
    - Advanced Free-Form Switches off the Expressions Editor entirely and modifies the Expressions list into a text area. In the text area, you can type the PCRE-format regular expression of your choice to define this policy. This is both the most powerful and the most difficult

method of creating a policy, and is recommended only for those thoroughly familiar with the NetScaler appliance and PCRE-formatregular expressions.
Caution: If you switch to Advanced Free Form expression editing mode, you cannot switch back to any of the other modes. Do not choose this expression editing mode unless you are sure that is what you want.

10. Repeat steps 6 through 8 to add any additional expressions you want to the Expressions list. You can mix named expressions and expressions created in the Expressions Editor. To the NetScaler appliance, they are all the same.
11. Click Create to create your new policy. Your new policy appears in the Policies pane list.
12. Click Close. To create additional Content Filtering policies, repeat the previous procedure. To remove a Content Filtering policy, select the policy in the Policies tab and click Remove.

Binding a content filtering policy

September 21, 2020

You must bind each content filtering policy to put it into effect. You can bind policies globally or to a particular virtual server. Globally bound policies are evaluated each time traffic directed to any virtual server matches the policy. Policies bound to a specific vserver are evaluated only when that vserver receives traffic that matches the policy.

Bind a policy to a virtual server by using the CLI

At the command prompt, type the following commands to bind a policy to a virtual server and verify the configuration:

```
bind lb vserver <name>@ -policyName <string> -priority <positive_integer>
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind lb vserver vs-loadbal -policyName policyTwo -priority 100
 Done
show lb vserver vs-loadbal
1)      vs-loadbal (10.102.29.20:80) - HTTP     Type: ADDRESS
        State: OUT OF SERVICE
        Last state change was at Wed Aug 19 09:05:47 2009 (+211 ms)
        Time since last state change: 2 days, 00:58:03.260
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

 Done
<!--NeedCopy-->
```

Globally bind a policy by using the CLI

At the command prompt, type the following commands to globally bind a policy and verify the configuration:

```
bind filter global (<policyName> [-priority <positive_integer>]) [-state ( ENABLED | DISABLED )]
show filter global
<!--NeedCopy-->
```

Example:

```
 bind filter global cf-pol -priority 1
 Done show filter global
1)      Policy Name: cf-pol     Priority: 1
 Done
<!--NeedCopy-->
```

Bind a policy to a virtual server by using the GUI

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server to which you want to bind the content filtering policy from the list, and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab, and then select the check box in the Active column of the filter policy that you want to bind to the virtual server.

4. Click OK. The policies you have bound display a check mark and the word Yes in the Policies Bound column of the Policies tab.

Globally bind a policy by using the GUI

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, in the Policies tab, select the policy that you want to bind, and then click Global Bindings.
3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select a policy, and then click Add. The policy is added to the Configured list.
   Note: To select multiple policies from the list, press and hold the Ctrl key, then click each policy you want.
4. Click OK, and then click Close. The policies you have bound display a check mark and the word Yes in the Globally Bound column of the Policies tab.

Configuring content filtering for a commonly used deployment scenario

September 21, 2020

This example provides instructions for using the configuration utility to implement a content filtering policy in which, if a requested URL contains root.exe or cmd.exe, the content filtering policy filter-CF-nimda is evaluated and the connection is reset.

To configure this content filtering policy, you must do the following:

- Enable content filtering
- Configure content filtering policy
- Bind content filtering policy globally or to a virtual server
- Verify the configuration

Note: Since this example uses a default content filtering action, you do not need to create a separate content filtering action.

Enable content filtering

1. In the navigation pane, expand System, and click Settings.
2. In the details pane, under Modes & Features, click Change Basic Features.
3. In the Configure Basic Features dialog box, select the Content Filtering check box, and then click OK.
4. In the Enable/Disable feature(s) dialog box, click Yes. A message appears in the status bar, stating that the selected feature is enabled.

Configure the content filtering policy filter-CF-nimda

1. Navigate to Security > Protection Features > Filter.
2. In the details pane, click Add. The Create Filter Policy dialog box appears.
3. In the Create Filter Policy dialog box, in the Filter Name text box, type the name filter-CF-nimda.
4. Select the Request Action option, and in the drop-down list, select RESET.
5. In the Expression frame, select Match Any Expression from the drop-down list, and then click Add.
6. In the Add Expression dialog box, Expression Type drop-down list, select General.
7. In the Flow Type drop-down list, select REQ.
8. In the Protocol drop-down list, select HTTP.
9. In the Qualifier drop-down list, select URL.
10. In the Operator drop-down list, select CONTAINS.
11. In the Value text box, type cmd.exe, and then click OK. The expression is added in the Expression text box.
12. To create another expression, repeat Steps 7 through 11, but in the Value text box, type root.exe. Then click OK, and finally click Close.
13. Click Create on the Create Filter Policy dialog box. The filter policy filter-CF-nimda appears in the Filter list.
14. Click Close.

Globally bind the content filtering policy

1. Navigate to Security > Protection Features > Filter. The Filter page appears in the right pane.
2. In the details pane, Policies tab, select the policy that you want to bind and click Global Bindings. The Bind/Unbind Filter Policies dialog box appears.
3. In the Bind/Unbind Filter Policies dialog box, in the Policy Name drop-down list, select the policy filter-CF-nimda, and click Add. The policy is added to the Configured list.
4. Click OK, and then click Close. The policy you have bound displays a check mark and Yes in the Globally Bound column of the Policies tab.

Bind the content filtering policy to a virtual server

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane virtual servers list, select vserver-CF-1 to which you want to bind the content filtering policy and click Open.
3. In the Configure Virtual Server (Load Balancing) dialog box, select the Policies tab.
4. In the Active column, select the check box for the policy filter-CF-nimda, and then click OK. Your content filtering policy is now active, and should be filtering requests. If it is functioning correctly, the Hits counter is incremented every time there is a request for a URL containing either

root.exe or cmd.exe. This allows you to confirm that your content filtering policy is working. The content filtering policy is bound to the virtual server.

Verify the content filtering configuration by using the command line interface

At the command prompt, type the following command to verify the content filtering configuration:

```
show filter policy filter-CF-nimda
<!--NeedCopy-->
```

Example:

```
sh filter policy filter-CF-nimda
        Name: filter-CF-nimda   Rule: REQ.HTTP.URL CONTAINS cmd.exe ||
            REQ.HTTP.URL CONTAINS root.exe
        Request action: RESET
        Response action:
        Hits: 0
Done
<!--NeedCopy-->
```

Note: The

Hits counter displays an integer that denotes the number of times the filter-CF-nimda policy is evaluated. In the preceding steps, the Hits counter is set to zero because no requests for a URL containing either cmd.exe or root.exe have been made yet. If you want to see the counter increment in real time, you can simply request a URL that contains either of these strings.

Verify the content filtering configuration by using the GUI

1. Navigate to **Security** > **Protection Features** > **Filter**.
2. In the details pane, select the filter policy filter-CF-nimda. The bottom of the pane should display the following:

```
    Request Action:**

    RESET

    Rule:

    REQ.HTTP.URL CONTAINS cmd.exe || REQ.HTTP.URL CONTAINS root.exe

    Hits:

     0
<!--NeedCopy-->
```

Troubleshooting

September 21, 2020

If the content filtering feature does not work as expected after you have configured it, you can use some common tools to access NetScaler resources and diagnose the problem.

Resources for troubleshooting

You can use the following tools and resources to troubleshoot most Content Filtering issues on a NetScaler appliance:

- The Wireshark application customized for the NetScaler trace files
- Trace files recorded when accessing the resource
- The configuration files
- The ns.log file
- The iehttpheaders, or a Fiddler trace or a similar utility

Troubleshooting content filtering issues

To troubleshoot a content filtering issue, proceed as follows:

- Verify that the feature is enabled.
- Verify that the content filtering policy is configured correctly. Pay special attention to the expression that evaluates the incoming requests.
  Note: Most content filtering issues are caused by incorrect configuration, and the error is most often in the policy configuration.
- Check the policy's Hits counter to verify that it is incrementing. If it is not, the policy is not getting evaluated.
- If the policy is getting evaluated and the required filtering is still not performed, you need to look into the policy expressions and action.
- If the policy's expression seems valid, test it by assigning a simple NSTRUE value to see if the evaluation of the expression is creating any issue.
- Reevaluate whether the filtering should be based on the request or the response.
- Verify that the action is configured correctly. For example, if a custom action is used to corrupt a header in the request, verify that the header name in the action is correct. If you are not sure about the header name, start a browser with iehttpheaders or a similar utility, and then verify

the headers in the request. When this feature is used, you can use nstrace to find out if appropriate action is performed when the packets leave NetScaler appliance.
- An iehttpheaders or Fiddler trace can help you find header options and names, client-side request headers, and response headers recorded on the client.
- To check the modifications made to the request header, record an nstrace on the NetScaler appliance or a Wireshark trace on the server.
- If none of the above measures resolves the issue, verify that the connection has not become untrackable, which can happen in certain circumstances. If a connection becomes untrackable, the appliance does not perform any application-level processing of the requests. In that event, contact Citrix Technical Support.

HTTP denial-of-service protection

September 21, 2020

Internet hackers can bring down a site by sending a surge of GET requests or other HTTP-level requests. HTTP Denial-of-Service (HTTP Dos) Protection provides an effective way to prevent such attacks from being relayed to your protected Web servers. The HTTP DoS feature also ensures that a NetScaler appliance located between the internet cloud and your Web servers is not brought down by an HTTP DoS attack.

Most attackers on the Internet use applications that discard responses to reduce computation costs, and minimize their size to avoid detection. The attackers focus on speed, devising ways to send attack packets, establish connections or send HTTP requests as rapidly as possible.

Real HTTP clients such as Internet Explorer, Firefox, or NetScape browsers can understand HTML Refresh meta tags, Java scripts, and cookies. In standard HTTP the clients have most of these features enabled. However, the dummy clients used in DoS attacks cannot parse the response from the server. If malicious clients attempt to parse and send requests intelligently, it becomes difficult for them to launch the attack aggressively.

When the NetScaler appliance detects an attack, it responds to a percentage of incoming requests with a Java or HTML script containing a simple refresh and cookie. (You configure that percentage by setting the Client Detect Rate parameter.) Real Web browsers and other Web-based client programs can parse this response and then resend a POST request with the cookie. DoS clients drop the NetScaler appliance’s response instead of parsing it, and their requests are therefore dropped as well.

Even when a legitimate client responds correctly to the NetScaler appliance’s refresh response, the cookie in the client’s POST request may become invalid in the following conditions:

- If the original request was made before the NetScaler appliance detected the DoS attack, but the resent request was made after the appliance had come under attack.

- When the client's think time exceeds four minutes, after which the cookie becomes invalid.

Both of these scenarios are rare, but not impossible. In addition, the HTTP DoS protection feature has the following limitations:

- Under an attack, all POST requests are dropped, and an error page with a cookie is sent.
- Under an attack, all embedded objects without a cookie are dropped, and an error page with a cookie is sent.

The HTTP DoS protection feature may affect other NetScaler features. Using DoS protection for a particular content switching policy, however, creates additional overhead because the policy engine must find the policy to be matched. There is some overhead for SSL requests due to SSL decryption of the encrypted data. Because most attacks are not on a secure network, though, the attack is less aggressive.

If you have implemented priority queuing, while it is under attack a NetScaler appliance places requests without proper cookies in a low-priority queue. Although this creates overhead, it protects your Web servers from false clients. HTTP DoS protection typically has minimal effect on throughput, since the test JavaScript is sent for a small percentage of requests only. The latency of requests is increased, because the client must re-issue the request after it receives the JavaScript. These requests are also queued

To implement HTTP DoS protection, you enable the feature and define a policy for applying this feature. Then you configure your services with the settings required for HTTP DoS. You also bind a TCP monitor to each service and bind your policy to each service to put it into effect.

Layer 3-4 SYN Denial-of-Service protection

September 21, 2020

Any NetScaler appliance with system software version 8.1 or later automatically provides protection against SYN DoS attacks.

To mount such an attack, a hacker initiates a large number of TCP connections but does not respond to the SYN-ACK messages sent by the victimized server. The source IP addresses in the SYN messages received by the server are typically spoofed. Because new SYN messages arrive before the half-open connections initiated by previous SYN messages time out, the number of such connections increases until the server no longer has enough memory available to accept new connections. In extreme cases, the system memory stack can overflow.

A NetScaler appliance defends against SYN flood attacks by using SYN cookies instead of maintaining half-open connections on the system memory stack. The appliance sends a cookie to each client that requests a TCP connection, but it does not maintain the states of half-open connections. Instead, the appliance allocates system memory for a connection only upon receiving the final ACK packet, or, for HTTP traffic, upon receiving an HTTP request. This prevents SYN attacks and allows normal TCP communications with legitimate clients to continue uninterrupted.

SYN DoS protection on the NetScaler appliance ensures the following:

- The memory of the NetScaler is not wasted on false SYN packets. Instead, memory is used to serve legitimate clients.
- Normal TCP communications with legitimate clients continue uninterrupted, even when the Web site is under SYN flood attack.

In addition, because the NetScaler appliance allocates memory for HTTP connection state only after it receives an HTTP request, it protects Web sites from idle connection attacks.

SYN DoS protection on your NetScaler appliance requires no external configuration. It is enabled by default.

Disable SYN Cookies

SYN cookies are enabled by default on a NetScaler appliance to prevent SYN attacks. If your deployment requires you to disable SYN cookies, for example, for server-initiated data connections or in cases where a connection is not established because the first packet is dropped or reordered, use one of the following methods to disable SYN cookies.

Disable SYN cookies by using the CLI

At the command prompt, type:

```
set nstcpprofile nstcp_default_profile -synCookie DISABLED

**Arguments**

synCookie

Enable or disable the SYNCOOKIE mechanism for TCP handshake with clients. Disabling SYNCOOKIE prevents SYN attack protection on the NetScaler appliance.

                Possible values: ENABLED, DISABLED

                Default: ENABLED
<!--NeedCopy-->
```

Disable SYN cookies by using the GUI

1. Navigate to **System** > **Profiles** > **TCP Profiles**.
2. Select a profile and click **Edit**.
3. Clear the **TCP SYN Cookie** check box.
4. Click **OK**.

Enable HTTP DoS protection

September 21, 2020

To configure HTTP DoS protection, you must first enable the feature.

Enable HTTP DoS protection by using the CLI

At the command prompt, type the following commands to enable HTTP DoS protection and verify the configuration:

```
enable ns feature HttpDoSProtection
show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature HttpDoSProtection
 Done
show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   OFF
 .
 .
 .
 10)    Global Server Load Balancing   GSLB                 ON
 11)    Http DoS Protection            HDOSP                ON
 12)    Content Filtering              CF                   ON
 .
 .
 23)    HTML Injection                 HTMLInjection        ON
 24)    NetScaler Push                 push                 OFF
 Done
```

```
<!--NeedCopy-->
```

Enable HTTP DoS protection by using the GUI

1. Navigate to System > Settings.
2. In the details pane, click Configure Advanced Features.
3. In the Configure Advanced Features dialog box, select the HTTP DoS Protection check box.
4. Click OK.

Define an HTTP DoS policy

September 21, 2020

After you enable HTTP DoS protection, you next create a policy.

Note: Before changing the default setting for Client Detect Rate, see Tuning the Client Detection/JavaScript Challenge Response Rate.

Configure a HTTP DoS policy by using the CLI

At the command prompt, type one of the following commands to configure an HTTP DoS policy and verify the configuration:

```
add dos policy <name> -qDepth <positive_integer> [-cltDetectRate <
    positive_integer>]
set dos policy <name> -qDepth <positive_integer> [-cltDetectRate <
    positive_integer>]
<!--NeedCopy-->
```

Example:

```
add dos policy pol-HTTP-DoS  -qDepth 30
 Done
set dos policy pol-HTTP-DoS  -qDepth 40
 Done
show dos policy
1)       Policy: pol-HTTP-DoS    QDepth: 40
 Done
<!--NeedCopy-->
```

Configure an HTTP DoS policy by using the GUI

1. Navigate to Security > Protection Features > HTTP DoS.
2. In the details pane, do one of the following:
   - To create a new policy, click Add.
   - To modify an existing policy, select the policy, and then click Open.
3. In the Create HTTP DoS Policy or Configure HTTP DoS Policy dialog box, specify values for the parameters:
   - Name*—name (You cannot change the name of an existing policy.)
   - QDepth*—qdepth
   - Client Detect Rate—cltDetectRate (Before changing the default setting for cltDetectRate, see Tuning the Client Detection/JavaScript Challenge Response Rate
4. Click OK to create your new policy. The policy that you created appears in the details pane, and the status bar displays a message indicating that the DoS policy is successfully configured.

Configuring an HTTP DoS service

September 21, 2020

After you configure an HTTP DoS policy, you must configure a service for your policy. The service accepts HTTP traffic that is protected by the HTTP DoS policy.

Configure an HTTP DoS service by using the CLI

At the command prompt, type one of the following commands to configure an HTTP DoS service and verify the configuration:

```
add service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED
set service <name>@ (<IP>@ | <serverName>@) HTTP <port> [-maxClient <positive_integer>] [-maxReq <positive_integer>] -state ENABLED
<!--NeedCopy-->
```

Example:

```
add service ser-HTTP-Dos1 10.102.29.40 HTTP 87
 Done
set service ser-HTTP-Dos1 -maxReq 20
 Done
show service
1)      srv-http-10 (10.102.29.30:80) - HTTP
```

```
        State: DOWN
        Last state change was at Wed Jul  8 07:49:52 2009
        Time since last state change: 34 days, 00:48:18.700
        Server Name: 10.102.29.30
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): NO
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
         .
         .
         .

5)      ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
        State: DOWN
        Last state change was at Tue Aug 11 08:23:40 2009
        Time since last state change: 0 days, 00:14:30.300
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
 Done
<!--NeedCopy-->
```

Configure an HTTP DoS service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, do one of the following:
   - To create a new service, click Add.
   - To modify an existing service, select the service, and then click Open.
3. In the Create Server or Configure Server dialog box, specify values for the following parameters, which correspond to the descriptions in "Parameters for configuring an HTTP DoS service" as follows (asterisk indicates a required parameter):
   - Service Name*—name (You cannot change the name of an existing service.)
   - Server*—IP or serverName (Specify one or the other, not both.)
   - Port*—port
4. If the Enable Service check box is not selected, select it.
5. Select the Advanced tab, and select the Override Global check box to enable those choices.
6. Specify values for the following parameters.
   - Max Clients*—maxClient
   - Max Requests*—maxReq
7. Click Create or OK, and then click Close. The service appears in the list of services.

Bind an HTTP DoS monitor and policy

September 21, 2020

To put HTTP DoS protection into effect after you have configured an HTTP DoS service, you must bind the monitor, and then bind the service to the HTTP DoS policy.

Bind the monitor to the service by using the CLI

At the command prompt, type the following commands to bind the monitor to the service and verify the configuration:

```
bind lb monitor <monitorName> <serviceName>
show lb monitor
<!--NeedCopy-->
```

Example:

```
bind lb monitor tcp ser-HTTP-DoS
 Done
> show lb monitor
```

```
1)    Name.......:   ping-default  Type......:      PING   State....ENABLED
2)    Name.......:    tcp-default  Type......:       TCP   State....ENABLED
3)    Name.......:           ping  Type......:      PING   State....ENABLED
4)    Name.......:            tcp  Type......:       TCP   State....ENABLED
5)    Name.......:           http  Type......:      HTTP   State....ENABLED
 .
 .
 .
17)    Name.......:       ldns-dns  Type......:  LDNS-DNS   State....ENABLED
 Done
<!--NeedCopy-->
```

Bind the policy to the service by using the CLI

At the command prompt, type the following commands to bind the policy to the service and verify the configuration:

```
bind service <serviceName> -policyName <policyname>
<!--NeedCopy-->
```

Example:

```
bind service ser-HTTP-DoS -policyName pol-HTTP-DoS
 Done
show service
1)        srv-http-10 (10.102.29.30:80) - HTTP
          State: DOWN
          Last state change was at Wed Jul  8 07:49:52 2009
          Time since last state change: 34 days, 01:24:58.510
          Server Name: 10.102.29.30
          Server ID : 0   Monitor Threshold : 0
          Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
          Use Source IP: NO
          Client Keepalive(CKA): NO
          Access Down Service: NO
          TCP Buffering(TCPB): NO
          HTTP Compression(CMP): NO
          Idle timeout: Client: 180 sec   Server: 360 sec
```

```
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED
        .
        .
        .
4)      ser-HTTP-Dos (10.102.29.18:88) - HTTP
        State: DOWN
        Last state change was at Tue Aug 11 08:19:45 2009
        Time since last state change: 0 days, 00:55:05.40
        Server Name: 10.102.29.18
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 0      Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: ON
        Down state flush: ENABLED
5)      ser-HTTP-Dos1 (10.102.29.40:87) - HTTP
        State: DOWN
        Last state change was at Tue Aug 11 08:23:40 2009
        Time since last state change: 0 days, 00:51:10.110
        Server Name: 10.102.29.40
        Server ID : 0   Monitor Threshold : 0
        Max Conn: 0     Max Req: 20     Max Bandwidth: 0 kbits
        Use Source IP: NO
        Client Keepalive(CKA): NO
        Access Down Service: NO
        TCP Buffering(TCPB): NO
        HTTP Compression(CMP): YES
        Idle timeout: Client: 180 sec   Server: 360 sec
        Client IP: DISABLED
        Cacheable: NO
        SC: OFF
        SP: OFF
        Down state flush: ENABLED
 Done
```

```
<!--NeedCopy-->
```

Bind the monitor and policy to the service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services.
2. In the details pane, select the service that you want to bind, and then click Open.
3. In the Configure Service dialog box, select the Monitor tab, click the name of the monitor you want in the Monitors list, and then click Add. The selected monitor is added to the Configured frame.
4. Select the Policies tab, then select the HTTP DoS tab.
5. Select a policy from the Available Policies list, and then click Add. The policy appears in the Configured Policies list.
6. Click OK, and then click Close. A message appears in the status bar, stating that the service has been configured.

Tuning the client detection/JavaScript challenger response rate

September 21, 2020

After you have enabled and configured HTTP DoS protection, if more than the maximum specified number of clients are waiting in the NetScaler surge queue for the HTTP DoS service, the HTTP DoS protection function is triggered. The default rate of challenged JavaScript responses sent to the client is one percent of the server response rate. The default response rate is inadequate in many real attack scenarios, however, and may need to be tuned.

For example, assume that the Web server is capable of a maximum of 500 responses/sec, but is receiving 10,000 Gets/sec. If 1% of the server responses are sent as JavaScript challenges, responses are reduced to almost none: 5 client (500 *0.01) JavaScript responses, for 10000 waiting client requests. Only about 0.05% of the real clients receive JavaScript challenge responses. However, if the client detection/JavaScript challenge response rate is very high (for example, 10%, generating 1000 challenge JavaScript responses per second), it may saturate the upstream links or harm the upstream network devices. Exercise care when modifying the default **Client Detect Rate** value.

If the configured triggering surge queue depth is, for example, 200, and the surge queue size is toggling between 199 and 200, the NetScaler toggles between the "attack" and "no-attack" modes, which is not desirable. The HTTP DoS feature includes a window mechanism with a default size of 20. After the surge queue size reaches the specified queue depth value, triggering "attack" mode, the surge queue size must fall to 20 less than the specified queue depth for the NetScaler appliance to enter "no-attack" mode. In the example, the surge queue size must fall below 180 before the appliance enters "no-attack" mode. During configuration, you must specify a value more than 20 for the **QDepth** parameter when adding a DoS policy or setting a DoS policy.

The triggering surge queue depth should be configured on the basis of previous observations of traffic characteristics. For more information about setting up a correct configuration, see Guidelines for HTTP DoS protection deployment.

Guidelines for HTTP DoS protection deployment

September 21, 2020

Citrix recommends you to deploy the HTTP DoS protection feature in a tested and planned manner and closely monitor its performance after the initial deployment. Use the following information to fine-tune the deployment of HTTP DoS Protection.

- The maximum number of concurrent connections supported by your servers.
- The average and normal values of the concurrent connections supported by your servers.
- The maximum output rate (responses/sec) that your server can generate.
- The average traffic that your server handles.
- The typical bandwidth of your network.
- The maximum bandwidth available upstream.
- The limits affecting bandwidth (such as external links, a particular router, or other critical devices on the path that may suffer from a traffic surge).
- Whether allowing a greater number of clients to connect is more important than protecting upstream network devices.

To determine the characteristics of a HTTP DoS attack, you should consider the following issues.

- What is the rate of incoming fake requests that you have experienced in the past?
- What types of requests have you received (complete posts, incomplete gets)?
- Did previous attacks saturate your downstream links? If not, what was the bandwidth?
- What types of source IP addresses and source ports did the HTTP requests have (e.g., IP addresses from one subnet, constant IP, ports increasing by one).
- What types of attacks do you expect in future? What type have you seen in the past?
- Any or all information that can help you tune DoS attack protection.

Priority queuing

September 21, 2020

The priority queuing feature lets you filter incoming HTTP traffic on the basis of categories that you create and define, and prioritize those HTTP requests accordingly. Priority queuing directs high-priority requests to the server ahead of low-priority requests, so that users who need resources for important business uses receive expedited access to your protected Web servers.

> **Note:**
>
> This feature is available with a NetScaler enterprise or platinum edition license.

To implement priority queuing, you create priority queuing policies that specify a priority, weight, threshold, and implicit action. When an incoming request matches a priority queuing policy, the request is processed as the associated action indicates. For example, you can create a priority queuing policy that places all matching requests above a certain threshold in a surge queue, while giving priority treatment to other requests.

You can bind up to three priority queuing policies to a single load balancing virtual server. The priority levels are:

- **Level 1:** A Level 1 policy processes priority requests.
- **Level 2:** A Level 2 policy processes requests that should receive responses as soon as Level 1 requests have been cleared from the queue.
- **Level 3:** A Level 3 policy processes non-priority requests that receive responses only after requests in the first two queues have been cleared.

You can use weighted queuing to adjust the relative priority of each of these queues. Weights can range from 0 to 101. A weight of 101 tells the NetScaler appliance to clear all requests in that queue before forwarding any requests in the lower-priority queues to the Web server. A weight of 0 tells the appliance to send requests in that queue to the Web server only when there are no requests waiting in any of the other queues.

You must assign a unique name to each priority queuing policy. Policy names can be up to 127 characters. Multiple policies bound to the same load balancing virtual server cannot have the same priority level. No two virtual servers that have one or more common underlying physical services can have priority queuing configured or enabled on both virtual servers simultaneously.

To configure priority queuing the NetScaler, you perform the following steps:

- Enable the load balancing feature
- Define a server and service
- Define a load balancing virtual server
- Bind the service to the load balancing virtual server
- Enable the priority queuing feature
- Create the priority queuing policies
- Bind the priority queuing policies to the load balancing virtual server
- Enable priority queuing on load balancing virtual server

For information about enabling load balancing, creating servers, creating virtual servers and services, and binding these servers and services, see the NetScaler Traffic Management Guide. For complete information about policies and expressions, see the NetScaler Policy Configuration and Reference Guide.

Enable priority queuing

September 21, 2020

To use the priority queuing feature the NetScaler appliance, you must first enable it.

Enable priority queuing by using the CLI

At the command prompt, type the following commands to enable priority queuing and verify the configuration:

```
enable ns feature PriorityQueuing
show ns feature
<!--NeedCopy-->
```

Example:

```
enable ns feature PriorityQueuing
 Done
show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   OFF
 3)     Load Balancing                 LB                   ON
 .
 .
 .
 8)     Priority Queuing               PQ                   ON
 .
 23)    HTML Injection                 HTMLInjection        ON
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

Enable priority queuing by using the GUI

1. Navigate to System > Settings.
2. In the details pane, click Configure advanced features.
3. In the Configure Advanced Features dialog box, select the Priority Queuing check box.
4. Click OK.

Configure a priority queuing policy

September 21, 2020

To configure a priority queuing policy, you can use either the configuration utility or the command line.

Note: For more information about using the command line, see "
Command Reference."

Configure a priority queuing policy by using the CLI

At the command prompt, type the following command to configure a priority queuing policy and verify the configuration:

```
add pq policy <policyName> -rule <expression> -priority <positive_integer> [-weight <positive_integer>] [-qDepth <positive_integer> | -polqDepth <positive_integer>]
<!--NeedCopy-->
```

Example:

```
add pq policy pol_cgibin -rule "URL == '/cgi-bin/'" -priority 1
 Done
show pq policy pol_cgibin
1)       Policy: pol_cgibin      Rule: URL == '/cgi-bin/'      Priority: 1     Weight: 10
        Hits: 0
 Done
<!--NeedCopy-->
```

Configure a priority queuing policy by using the GUI

1. Navigate to Security > Protection Features > Priority Queuing.

2. In the details pane, do one of the following:
    - To create a new policy, click Add.
    - To modify an existing policy, select the policy, and then click Open.
3. If you are creating a new policy, in the Create PQ Policy dialog box, in the Name text box, type a name for your new policy.

    The name can consist of from one to 127 letters, numbers and the hyphen and underscore symbol.

    If you are modifying an existing policy, skip this step. You cannot change the name of an existing policy.
4. In the Rule text box, either enter the policy expression directly, or click New to create a policy expression. If you click New, perform the following steps:
    a) In the Create Expression dialog box, click Add.

    b) In the Add Expression dialog box, leave Expression Type set to General, and in the Flow Type drop-down list, select a Flow Type. Your choices are REQ (for requests) and RES (for responses).

    c) In the Protocol drop-down list, select a protocol. If you selected REQ in the previous step, your choices are HTTP (Web-based connections), SSL (secure Web connections), TCP and IP. If you selected RES in the previous step, your choices are HTTP, TCP and IP.

    d) In the Qualifier drop-down list, select a qualifier.

    Your choices depend upon your selections in the previous step. Common choices are HTTP VERSION (the version of the HTTP connection), HTTP HEADER (the specified HTTP header), TCP SOURCEPORT/ DESTPORT (the source or destination port of a TCP connection), and IP SOURCEIP/DESTIP (the source or destination IP of the connection).

    If you choose HTTP HEADER, the Header text box appears beneath the original row of text boxes. You fill in the name of the HTTP header you want.

    For a complete description of the available choices, see "Policies and Expressions."

    e) In the Operator drop-down list, select an operator.

    For a complete description of the available choices, see "Policies and Expressions."

    f) In the Value text box, type the value you want to test for.

    This may be a text string or a number, depending upon the context. For a complete description of values appropriate to the specific context, see "Policies and Expressions."

    g) Click OK. The expression is added in the Expression text box.

    h) Click Create. The expression appears in the Rule text box.

5. In the Priority and Weight text boxes, type numeric values, for example, 1 and 30. For more information about Priority and Weight, see "Set up weighted queuing."
6. Enter a numeric value for either Queue Depth or Policy Queue Depth, for example 234, and click Create.
    - Queue Depth Defines the total number of waiting clients or requests on the virtual server to which the policy is bound.
    - Policy Queue Depth Defines the total number of waiting clients or requests belonging to the policy.

    The policy is created and appears in the Priority Queuing page.

    Note: To create additional priority queuing policies, repeat the procedure in the preceding section, and click
    Close after you finish.

Bind a priority queuing policy

September 21, 2020

After you create a priority queuing policy, you must bind it to the appropriate virtual server to put it into effect.

Bind a priority queuing policy by using the CLI

At the command prompt, type the following commands to bind a policy and verify the configuration:

```
bind lb vserver <name> -policyName <policyname>
show lb vserver <name>
<!--NeedCopy-->
```

Example:

```
bind lb vserver lbvip -policyname pol_cgibin
 Done
show lb vserver lbvip
        lbvip (8.7.6.6:80) - HTTP       Type: ADDRESS
        State: DOWN
        Last state change was at Wed Jul 15 05:54:24 2009 (+782 ms)
        Time since last state change: 26 days, 05:44:37.370
        Effective State: DOWN
        Client Idle Timeout: 180 sec
        Down state flush: ENABLED
        Disable Primary Vserver On Down : DISABLED
        Port Rewrite : DISABLED
        No. of Bound Services :  0 (Total)       0 (Active)
        Configured Method: LEASTCONNECTION
        Mode: IP
        Persistence: NONE
        Vserver IP and Port insertion: OFF
        Push: DISABLED  Push VServer:
        Push Multi Clients: NO
        Push Label Rule: none

1)      Policy : ns_cmp_msapp Priority:0

1)      Priority Queuing Policy : pol_cgibin
 Done
<!--NeedCopy-->
```

Bind a priority queuing policy by using the GUI

1. In the navigation pane, locate and select the virtual server to which you want to bind the priority queuing policy.
    - To select a load balancing virtual server, expand Traffic Management > Load Balancing > Virtual Servers, then select the load balancing virtual server that you want..
    - To select a content switching virtual server, expand Traffic Management > Content Switching > Virtual Servers, then select the content switching virtual server that you want..
2. In the Configure Virtual Server dialog box, select the Policies tab.
3. Click the double right-arrow (») symbol to display the complete list of policy types, and then select Priority Queuing from the drop-down list.
4. Click Insert Policy.
5. In the Policy Name row, select the policy that you want to bind from the drop-down list.
6. ClickOK to save your changes.

Set up weighted queuing

September 21, 2020

When priority queuing is implemented, lower-priority requests are typically kept on hold while higher-priority requests are served. The lower-priority requests may therefore be delayed if there is a constant flow of higher-priority requests.

To prevent delays for low-priority requests across multiple priority levels, you can configure weighted queuing for serving requests. The default weights for the priorities are:

- Gold - Priority 1 - Weight 3
- Silver - Priority 2 - Weight 2
- Bronze - Priority 3 - Weight 1

You assign the minimum weight, zero (0), to requests that the NetScaler appliance should send to the server only if no requests are stored in any of the other queues. You assign the maximum weight, 101, to requests that the appliance should send to the server immediately, ahead of any requests stored in any of the other queues. Weights between these two set the relative priority of a particular queue in relation to the other queues. Queues with a higher weight are processed first; queues with a lower weight after the others have been processed. To assign the weights, see "Configure a priority queuing policy."

Note: The weight assigned to a higher-priority queue must be larger than the weight assigned to a lower-priority queue. For example, the weight assigned to The Gold (Priority 1) queue must be greater than the weight assigned to the Silver (Priority 2) queue.

SureConnect

September 21, 2020

You can use the SureConnect feature of the NetScaler appliance to service all incoming connections with either the requested content or a custom Web page that displays information about a delay in the request being serviced.

When servers are overloaded with the requests, the servers might either respond slowly or not at all. The SureConnect feature enables the NetScaler appliance to detect and compensate such conditions by ensuring that every client request gets serviced in some way, such as either a custom Web page or actual content is sent to the client.

SureConnect is activated when the response time or maximum server connections to a client request exceeds a limit that you have set. The SureConnect browser window displays one of the following:

- A progress bar with the amount of time remaining until the requested content will be available.
- Alternate Web content of your choice (alternate page).
- Both a progress bar and alternate page.
- Complete custom content of your choice.

You can configure whether the SureConnect progress bar alone is displayed or both the progress bar and the alternate page are displayed.

When the server becomes responsive again, the original request for content is served. If the user chooses, the alternate content window can remain in focus.

Subsequent requests from the same user within the same session are served immediately. This can be configured using the settings described later in this section.

SureConnect can be activated when a response is delayed, and when the number of user connections to a given URL exceeds a specified threshold.

SureConnect works with all standard browsers, including Microsoft Internet Explorer, Netscape Navigator, and Mozilla Firefox.

SureConnect is advantageous in the following situations:

- **Full server queue:** The server can respond fast, but there are too many users. This results in the server's queue being full and unable to process additional client requests.

  SureConnect Solution: In this situation, the SureConnect window is displayed, showing the time left until the content will be available. The alternate page is displayed under the progress bar, if an alternate page has been configured.

- **Large response delay:** The server response is slow. Typically, if a Web server does not respond to a client request quickly, the user will leave the site.

  SureConnect Solution: When the predicted delay reaches a configured time threshold, the SureConnect window displays the progress bar and the optional alternate page in the client browser.

- **Client time-out:** When the client requests content from a very slow Web site, a time-out message displays in the client browser, and the content is not delivered. The user may leave the site.

  SureConnect Solution: The appliance stores the request until the server is no longer busy and delivers the requested content to the client.

- **Server experiencing a traffic surge:** The server typically responds quickly, but the current load of open connections is greater than the server capacity to serve them. Therefore, the server response is delayed.

  SureConnect Solution: A SureConnect window is displayed in the client browser, showing the time left. The alternate page from the server is also displayed if it has been configured.

Installing SureConnect

September 21, 2020

SureConnect files must be installed on the alternate content server, which can be the same as the primary server.

On a Windows server, extract the sc_xx.exe file (where xx is the build number), or on a UNIX server, extract the sc_xx.tar file (where xx is the build number).

Note: You must install SureConnect in the default Web root directory.

If the alternate content server is the same as the primary server, place the SureConnect and alternate content files in any directory under the Web root directory. Specify this path when you add a policy to configure SureConnect. By default, SureConnect files are installed in the /NetScaler appliance directory under the default Web root directory.

If the alternate content server is different than the primary server, the SureConnect and alternate content files must be in a unique directory under the Web root directory. By default, this unique directory is the /NetScaler system directory. Specify this path when you add a policy to configure SureConnect.

The following files are extracted:

- Alternate content files (progressbar.htm, alternatepage.htm, and barandpage.htm)
- System-Logo.png
- Customer-Logo.png
- Sample.png
- README.txt.

Installing on UNIX

This section describes how to install SureConnect alternate content on a UNIX server. The following are the prerequisites:

- The UNIX server is running the Apache server.
- The shell with the # prompt is in use.
- Apache is installed in the default location.
- The sc_xx.tar file is downloaded from the organization's Web site into the /var/ftp/incoming directory.

To install SureConnect

1. At the command prompt, navigate to the htdocs directory:

   ```
   cd /usr/local/apache/htdocs
   <!--NeedCopy-->
   ```

2. Type the following command:

```
tar xvpf/var/ftp/incoming/sc_xx.tar
<!--NeedCopy-->
```

The output from the
.tar file is displayed. A
/NetScaler system directory is created under the specified path and the SureConnect files are installed.

Installing on Windows

This section describes how to install SureConnect alternate content on a Windows server. The following are the prerequisites:

- The server is running the Microsoft Internet Information Server.
- The DOS prompt is being used.
- The SureConnect zip (self-extracting) file is downloaded from the organization Web site using FTP into the C:inetpubwwwroot directory.

To install SureConnect on Windows

Do one of the following:

- At the command prompt, navigate to the wwwroot directory:

  ```
  cd c:inetpubwwwroot
  <!--NeedCopy-->
  ```

- Type the name of the executable file:

  ```
  sc_xx.exe
  <!--NeedCopy-->
  ```

- Double-click the sc_xx.exe icon from the Microsoft Windows Explorer Web browser, extract from the compressed file into the default path (for example, the c:inetpubwwwroot directory).

Output from the
zip file is displayed. A
/NetScaler system directory is created under the specified path, and the SureConnect files are installed.

Configure SureConnect

September 21, 2020

The following topics describe how to configure SureConnect for scenarios involving alternate server failure.

Configure the response for alternate server failure

If the alternate server fails, and the primary server cannot immediately deliver the requested content to the client, SureConnect does not display alternate content from the failed alternate server in the client Web browser.

The NetScaler appliance automatically sends a response to the client browser. You can customize the server response to display information suited to your needs.

The default response is:

Your Request is being processed… Estimated Time: __ Secs

Customize the default response

The NetScaler appliance automatically sends the response to the client if the alternate server fails, or if the appliance is configured to send the default response.

To customize the default response of the appliance, create a vsr.htm file (a sample is provided in this section) as follows:

- The file can contain any valid HTML statements other than embedded objects.
- The file size cannot exceed 800 bytes.
- The file must reside on the NetScaler appliance. If you have a high availability (HA) setup, the file must reside on the primary and secondary nodes. Any changes made to the file on the primary node must also be applied to the file on the secondary node.
- Put vsr.htm file in the /etc directory.

To customize the default response

Change any of the contents between the </HEAD> and </HTML> tags in the vsr.htm file. Following is the sample content from vsr.htm file. The sections that you can edit are in bold text.

```
HTTP/1.1 200 OK
Server: NS_WS3.0
Content-Type: text/html
Cache-control: no-cache
Pragma: no-cache
Set-Cookie: NSC_BPIP=@@SID@@; path=/
<HTML> <HEAD> <META HTTP-EQUIV="Refresh" CONTENT="0">
</HEAD> <font color=blue size=5>Your request is being processed...
<br>Estimated Delay: @@DELAY@@ Sec </font> </HTML>
```

```
<!--NeedCopy-->
```

Note: Include @@DELAY@@ to display the predicted delayed response time in seconds.

SureConnect with in-memory response (NS action)

When defining the SureConnect policy by using the add sc policy command, you can configure the NetScaler appliance to serve alternative content to the client.

To enable SureConnect and configure the in-memory response, perform the following tasks:

- Enable the SureConnect feature on the appliance by using the enable feature SC command
- Define the services by using the `add service <servicename> <IP address> <servicetype> <port>` command. This identifies the original server for which the SureConnect is configured and the types of services.
- Add a SureConnect policy by using the add sc policy command. You can configure a URL-based policy or a rule-based policy. The incoming requests are validated against the URL or rule you specify in the policy.

Note: You can configure the SureConnect feature on a load balancing virtual server. In that case, perform the following additional actions:

- Enable Load Balancing by using the enable feature LB command.
- Enable SureConnect feature on the virtual server by using the `set lb vserver <vservername> -sc ON` command.
- Bind services to the virtual server by using the `bind lb vserver <name> <serviceName>` command.
- Bind policies to the virtual server by using the `bind lb vserver <name> -policyname <name>` command.

The following example illustrates how to configure SureConnect for the load balancing feature so that SureConnect will display alternative content from the NetScaler appliance.

In this example, two physical servers, with IP addresses, 10.101.3.187 and 10.101.3.188 are load balanced by the NetScaler appliance. The appliance has one configured virtual server, vs-NSact, whose IP address is 10.101.3.201. The file that contains the alternative content is vsr.htm. It is copied from the file system into system memory. Services are loaded until the SureConnect policy triggers, and the appliance supplies the alternate content.

```
enable feature SC LB
add service psvc1 10.101.3.187 http 80
add service psvc2 10.101.3.188 http 80
add lb vserver vs-NSact HTTP 10.101.3.201 80
bind lb vserver vs-NSact psvc1
bind lb vserver vs-NSact psvc2
```

```
add sc policy policyNS -url /cgi-bin/*.cgi -delay 400000
-action NS
set sc parameter -vsr /nsconfig/ssl/vsr.htm
bind lb vserver vs-NSact -policyName policyNS
set lb vserver vs-NSact -sc ON
save config
<!--NeedCopy-->
```

Table 1. Parameter values used in this example

Service	
Name	psvc1, psvc2
Server	10.101.3.187, 10.101.3.188
Protocol	HTTP
Port	80
Load Balancing Virtual Server	
Name	vs-NSact
IP Address	10.101.3.201
Protocol	HTTP
Port	80
SureConnect Policy	
Name	policyNS
URL	/cgi-bin/*.cgi
Delay(microseconds)	400000
SC Parameter	
VSR File Name	vsr.htm

To configure this example by using the GUI

1. In the In the navigation pane, navigate to System > Settings. In the Modes and Features pane, perform the following actions:
    a) Click Configure Basic Features, select Load Balancing, and Click Go.
    b) Click Configure Advanced Features, select SureConnect, and Click Go.
2. In the navigation pane, navigate to Security > Protection Features > SureConnect. In the details

pane, click Parameters. In the Configure SureConnect Parameters window, browse and select the VSR filename.

3. Navigate to Traffic Management > Load Balancing > Services. In the details pane, click Add. In the **Create Services** window, enter the paramter values as shown in Table 5-1, and click **OK**.
4. Navigate to Traffic Management > Load Balancing > Virtual servers. In the details pane, click Add. In the Create Virtual Server (Load Balancing) dialog box, enter the values shown in Table 5.1 for the Load Balancing Virtual Server parameters and click OK.
5. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. The Configure Virtual system (Load Balancing) dialog box, displays the list of configured services. Select services psvc1 and psvc2 and click OK.
6. In the navigation pane, expand Security > Protection Features > SureConnect. In the details pane, click Add. Create the policy with the values as given in the parameters table.
7. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, click the Policies tab. Click » to expand the features. Select **SureConnect**. When the list of SureConnect polices appear, select policyNS and click OK.
8. In the navigation pane, navigate to Traffic Management > Load Balancing > Virtual servers. Select the virtual server vs-NSact and click Open in the details pane. In the Configure Virtual system (Load Balancing) dialog box, on the Advanced tab, select SC and click OK.

Configure the SureConnect policies

You can configure the following SureConnect policies. The NetScaler appliance matches incoming requests in the order the policies are configured:

- Exact URL-based policies
- Wildcard rule-based policies

Configure exact URL based policies

When you configure an exact URL based policy, the NetScaler appliance matches the incoming request against the URL that has been configured in the policy. URL based policies take precedence over rule based policies.

Configure an exact URL based policy by using the CLI

At the command prompt, type:

```
add sc policy <name> [-url <URL> | -rule <expression>] [ -delay <usecs>] [ -maxConn <positive_integer>] [ -action (ACS <altContentSvcName> <altContentPath>) | NS | NOACTION)]
```

```
<!--NeedCopy-->
```

Configure an exact URL based policy by using the GUI

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, set the following parameters:
    - Name*
    - URL (Make sure that the URL check box is selected)
    - Value*
    - Delay (microseconds)*
    - Maximum Client Connections
    - Action (Select from the Choose Action list.)
    - Alternate Service Name (if you select ACS as the Action)
    - Alternate Content Path (if you select ACS as the Action)

    *A required parameter
4. Click Create, and click Close. The URL based policy appears in the right pane, and a message displays in the status bar that the policy is successfully configured.

Configure wildcard rule-based policies

SureConnect matches the incoming requests to a defined rule, if you configure a rule-based policy.

Configure a SureConnect policy based on a wildcard rule by using the CLI

1. Create the expression(s).

    Use the add expression command to create each expression.

2. Create the rule(s).

    Use the add sc policy command with the -rule expression_logic argument to specify the rule(s). In the -rule expression_logic argument, refer to the expression(s) you created in step 1.

    Repeat this command to create and name each rule.

The following example creates a rule "rule = = /*.cgi":

```
add vserver vs-lb http 1.1.1.1 80
add expression expr1 url == /cgi-bin/*.cgi
add expression expr2 url == /index.html
add sc policy surecpolicy1 -rule (expr1||expr2) -delay 1000000 -action
   NS
bind lb vserver vs-lb -policyName surecpolicy1
```

```
<!--NeedCopy-->
```

To complete the SureConnect configuration, you will need to enter additional commands, beyond those shown in the example.

Configure a wildcard rule-based policy by using the GUI

1. Navigate to Security > Protection Features > SureConnect.
2. In the details pane, click Add.
3. In the Create SureConnect Policy dialog box, in the Name text box, type the name of the policy.
4. Under What to Monitor, click Expression, and then click Configure.
5. In the Create Expression dialog box, click Add.
6. In the Add Expression dialog box, enter an expression. For example, you can select an Expression Type of General, a Flow Type of REQ, a Protocol of HTTP, a Qualifier of URLQUERY, an Operator of CONTAINS, and in the Value text box, type AA. For more information about expressions, see "Policies and Expressions."
7. Click OK, and click Close.
8. In the Create Expression dialog box, click Create.

Examples of wildcard rules:

"/sports/*" matches all URLs under /sports

"/sports*" matches all URLs whose prefix matches "/sports", starting at the beginning of the URL.

"/*.jsp" matches all URLs whose file extension is "jsp"

When configuring rule-based policies, first add the more specific rule-based policies, before adding more generic rules (for example, add /cgi-bin/sports.*cgi before adding /cgi-bin/*.cgi).

Display the configured SureConnect policy

To view the SureConnect policy that you have configured, at the NetScaler command prompt, enter the
show sc policy command.

Customize the alternate content file

When SureConnect activates, it can display alternate content from one of the following files that you have configured:

- **progressbar.htm.** Displays the progress information.
- **alternatepage.htm.** Displays an alternate page.
- **barandpage.htm.** Displays both the progress information and an alternate page.

The alternate content files are JavaScript files. During SureConnect installation, these files are copied onto the server that contains the alternate content. These files can contain alternate content (including an alternate page) or references to other files that contain the alternate content.

This section describes the changes you can make to the alternate content file provided by the appliance.

```
//**** DEFINE YOUR VALUES HERE ****
var alt_url = "/NetScaler system /sample.png";
var alt_url = "http://www.DomainName.com";
var NetScaler system _logo = "NetScaler_logo.png";
var our_logo = "NetScaler_logo.png";
var height = 450;
var width = 550;
var top = 200;
var left = 200;
var popunder = "no"; //specify yes for pop-under & no for pop-up
var shift_focus = "yes" //if you want to send pop-up to background on
    getting primary content else specify no
//**** YOUR DEFINITIONS ENDS HERE ****
<!--NeedCopy-->
```

You can make these changes:

- **var alt_url.** Specify the URL for the alternate content if a file provides the alternate content. For example:

  `pre codeblock var alt_url = “/NetScaler system/sports.htm” <!--NeedCopy -->`

  Note: The alternate content file must be present in the /NetScaler system directory under the documents root of the Web server.

- **var our_logo.** Specify the image file of your organization logo.
- **var height.** Specify the height of the SureConnect window.
- **var width.** Specify the width of the SureConnect window.
- **var top and var left.** Specify the position of the SureConnect window.
- **var popunder.** Specifies the position of the alternate content window. Specify the value as NO to place the alternate content window above the original window. Specify the value as YES to place the alternate content window beneath the original window.
- **var shift_focus.** Specify the focus of the alternate content window. YES places the pop-up window in the background when getting the primary content. NO always keeps the pop-up window in focus, even when getting the primary content.

Note: For more information, see the
README.txt file provided by the appliance with other alternate content files.

Configure SureConnect for NetScaler features

This section describes how SureConnect works in combination with the load balancing, content switching, cache redirection, and high availability features of the NetScaler appliance.

Configure SureConnect for load balancing

You can use SureConnect in environments where the primary servers use the load balancing feature, with or without alternate servers. If the load balancing virtual server configured for SureConnect fails, the backup virtual server (if there is one) handles the traffic. Backup virtual servers do not support SureConnect policies.

Note: For information about load balancing, see
Load Balancing.

Configure SureConnect for cache redirection

You can use SureConnect in environments where cache redirection is configured. The primary server is a load balancing virtual server bound to the cache redirection virtual server. Regardless of any rules configured for the cache redirection feature:

- You can configure any URL for SureConnect.
- Once SureConnect is activated for a client, requests from the client are always sent to the origin server.

Configure SureConnect for high availability

SureConnect is compatible with NetScaler appliances operating in high availability mode.

Note: If the optional
vsr.htm file is used, it must be present in both nodes (primary and secondary) and must use the same name and directory.

Activating SureConnect

September 21, 2020

You can set the NetScaler appliance to activate SureConnect if either of two criteria match. Both criteria are arguments to the
add sc policy command, as described here:

- -delay <microseconds>

  The first time the client requests the URL, the appliance records how long the server takes to respond. The appliance will not activate SureConnect until the second time the URL is requested. The first and second requests may be from the same or different clients.

  If you set -delay argument, SureConnect will be activated the second time the delay reaches the threshold you set.

  </div>

- -maxConn

  When the appliance receives a request, it checks the number of connections to the server for the configured URL. SureConnect is activated if the number of connections is greater than or equal to the value that you set for the -maxConn argument.

  If you will be providing alternate content to be displayed in the client's Web browser, you should configure the -action argument of the add sc policy command. This specifies for the NetScaler appliance whether the alternate content is coming from a dedicated alternate server (-action ACS) or the appliance (-action NS).

  When SureConnect is activated by the –maxConn argument, the SureConnect window and progress bar are displayed in the client's browser (with an alternate page, if configured).

SureConnect environments

September 21, 2020

The following topics describe SureConnect environments.

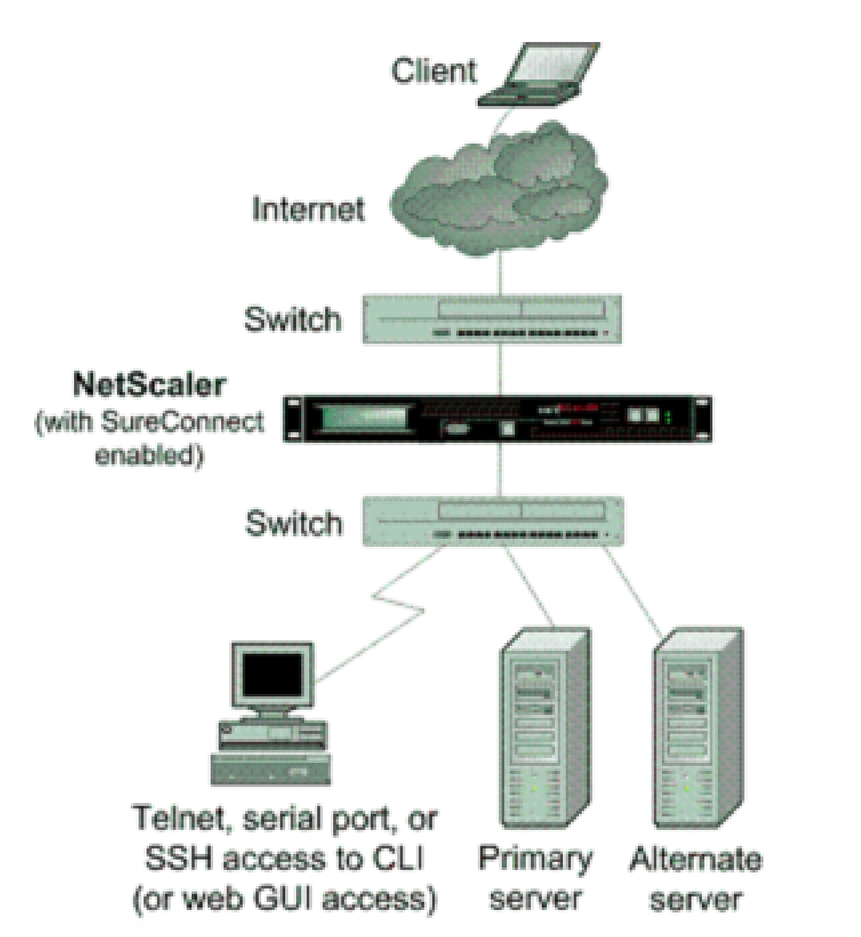
Primary and alternate servers

The SureConnect environment uses a dedicated server to provide alternate content when the requested content is not available. The alternate content may include an alternate page, plus optional components such as frame set, organization logo, and so on. The alternate and primary servers can be the same server.

You can configure SureConnect to display a progress bar when the requested content is not available (or the progress bar and an alternate page).

The following figure illustrates the SureConnect environment.

Figure 1. SureConnect - Primary and Alternate Servers


Client
Internet
Switch
NetScaler
(with SureConnect enabled)
Switch
Telnet, serial port, or SSH access to CLI (or web GUI access)
Primary server
Alternate server

Configuration checklist

Complete the following checklist before you start configuration:

- The same builds are running for the appliance and for the SureConnect files as suggested by appliance staff.
  - Appliance Build Number: ________
  - SureConnect (sc_xx.exe) Build Number: _______
- The latest SureConnect files (style files) are extracted to:
  - All primary servers (required for NS action.
  - The alternate content server (required for ACS action).
- All customizations to the latest style and vsr.htm files are applied.
- The alternate content server is accessible from the Internet (required for ACS action).

- If the -redirectURL URL argument of the add vserver CLI command needs to be specified:
    - The URL is up and running.
    - This URL is not on the configured servers.
    - This URL does not match any content in the vserver (that is, do not redirect a missing URL to itself). Redirecting a missing URL to itself can send some browsers into an infinite loop.
- All URLs to be configured for SureConnect are top-level URLs only. (Only the URLs that occupy the whole window or frame can be configured, not the embedded objects).

Following are the steps to configure SureConnect in a setup with a primary server and a dedicated alternate server:

- Enable the SureConnect feature
- Add the SureConnect policy
- Bind the SureConnect policy

You can optionally configure the following:

- Redirect the client to another URL if the primary server fails, or send a customized response to the client if the alternate server fails.
- If the servers do not provide alternate content, send a default or customized response.

Redirect the client to another URL

1. Enable the SureConnect feature.
2. Define the primary server and its service.

   You must identify the original server for which SureConnect support is being configured. At the NetScaler command prompt, type the following command:

   ```
   add service <serviceName> <IP> HTTP <port>
   <!--NeedCopy-->
   ```

   where `<serviceName>` assigns a name for the service; `<IP>` is the server's IP address; and `<port>` is the port number that the service will use.

   Repeat use of the add service CLI command for each service that is to be added.

   You can also configure SureConnect on a load balancing virtual server. At the NetScaler command prompt, type the following command:

   ```
   add vserver <name> HTTP <IP> <port>
   <!--NeedCopy-->
   ```

3. Define and bind the SureConnect policy as follows. If you are configuring a rule-based policy, perform this step as described in "Configure wildcard rule-based policies." To configure a URL-based policy, at the NetScaler command prompt, type the following command:

   ```
   add sc policy <name> [-url <URL>] [-delay <microsec>] [-maxConn <
       positiveInteger>]
   <!--NeedCopy-->
   ```

   For a detailed description of the add sc policy command, see "[Command Reference]https://developer-docs.citrix.com/projects/netscaler-command-reference/en/latest/."

   To bind the SureConnect policy, at the NetScaler command prompt, type the following command:

   ```
   bind service <serviceName> -policyname <string>
   <!--NeedCopy-->
   ```

   where `<serviceName>` is the name of the service defined in step 2, and `<string>` is the name of the SureConnect policy.

   Repeat the bind service command for each policy created.

   You must include the alternate content page in the altContSvcName argument, and in the altContPath argument of the add sc policy command.

   In the following example, the name of the alternate content file is /NetScaler system /barandpage.htm, and this file resides in svc2.

4. To save the configuration, at the NetScaler command prompt, type the following command:

   ```
   save config
   <!--NeedCopy-->
   ```

Example configurations

The following examples illustrate various SureConnect configurations.

The examples assume that monitoring of physical services is enabled. If the alternate system is down, SureConnect will deliver the alternate content from the system itself.

Example 1 - SureConnect progress bar and alternate page

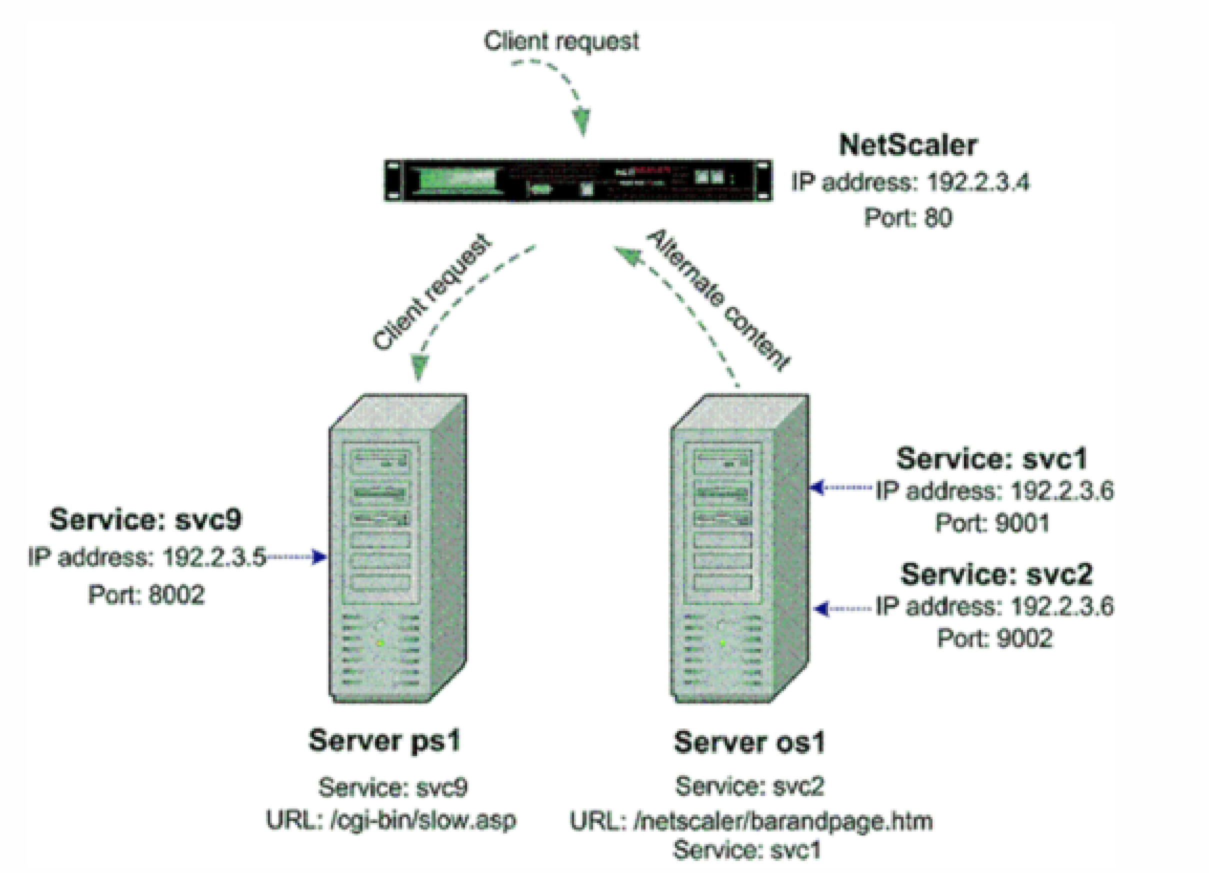
You can configure SureConnect to display both the progress bar and an alternate page to the user.

To bind a SureConnect policy to a load balancing virtual server, at the command prompt, type the following commands:

```
bind lb vserver <virtualServerName> -policyName <string>
<!--NeedCopy-->
```

where `<virtualServerName>` is the name of the load balancing virtual server defined in step 2 of the configuration process, and `<string>` is the name of the SureConnect policy defined in step 3.

Figure 2. SureConnect Configuration - Example 1


Client request
NetScaler
IP address: 192.2.3.4
Port: 80
Client request
Alternate content
Service: svc9
IP address: 192.2.3.5
Port: 8002
Service: svc1
IP address: 192.2.3.6
Port: 9001
Service: svc2
IP address: 192.2.3.6
Port: 9002
Server ps1
Service: svc9
URL: /cgi-bin/slow.asp
Server os1
Service: svc2
URL: /netscaler/barandpage.htm
Service: svc1

At the NetScaler command prompt, type the following commands:

```
enable feature SC
show ns info
add service svc2 192.2.3.6 HTTP 9002
show server
show service svc2
add service svc9 192.2.3.5 HTTP 8002
add sc policy policy8 -url /cgi-bin/slow.asp
-delay 3000000 -action ACS svc2 /NetScaler 9000 system barandpage.htm
bind service svc9 -policyname policy8
set service svc9 -sc ON
save config
```

```
<!--NeedCopy-->
```

After you configure SureConnect, you can enter commands that show information to verify what you have configured.

Example 2 - SureConnect progress bar only

In this example, SureConnect will display only the progress bar. The server orgsrvr with IP address 10.101.8.187 has service orgsvc. This server is connected to the appliance. The service is bound to the appliance. The progressbar.htm file specifies that only the progress bar will be displayed.

At the NetScaler command prompt, type the following commands:

```
enable feature SC
add service orgsvc 10.101.3.187 HTTP 80
add sc policy policy9 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS orgsvc /NetScaler 9000 system / progressbar.
    htm
bind service orgsvc -policyname policy9
set service orgsvc -sc ON
save config
<!--NeedCopy-->
```

Example 3 - SureConnect with load balancing

This example illustrates how to configure the load balancing feature so that SureConnect will display alternate contents from the primary server. For information about load balancing, see Load Balancing.

In this example, two physical servers with IP 10.101.3.187 and 10.101.3.188 are being load balanced by the appliance. The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on both servers.

The appliance has one configured virtual server address: 10.101.3.201. At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add vserver vs-SureC HTTP 10.101.3.201 80
bind lb vserver vs-SureC psvc1
bind lb vserver vs-SureC psvc2
add sc policy policy9 -url /cgi-bin/slow.asp -delay 4000000
-action ACS vs-SureC /NetScaler system /alternatepage.htm
```

```
bind lb vserver vs-SureC -policyName policy9
set lb vserver vs-SureC -sc ON
save config
<!--NeedCopy-->
```

Example 4 - SureConnect with load balancing (ACS action)

This example illustrates how to configure the NetScaler appliance load balancing feature so that SureConnect will display alternate content from the alternate server. For information about load balancing, see Load Balancing.

In this case, there are two physical servers, IP 10.101.3.187 and 10.101.3.188. Both are being load balanced by the appliance.

The name and location of the alternate page file are specified in file barandpage.htm, which resides on a third server not being load balanced.

The third server’s IP address is 10.101.3.189. Because barandpage.htm is specified, the progress bar and alternate page will both be displayed.

The appliance has one configured virtual server “vsvr” whose IP address (Virtual Server) is 10.101.3.200.

At the NetScaler command prompt, type the following commands:

```
enable feature SC LB
add service psvc1 10.101.3.187 HTTP 80
add service psvc2 10.101.3.188 HTTP 80
add service alt-cont-svc 10.101.3.189 HTTP 80
add vserver vsvr HTTP 10.101.3.200 80
bind lb vserver vsvr psvc1
bind lb vserver vsvr psvc2
add sc policy policy10 -url /cgi-bin/slow.asp
-delay 4000000 -action ACS alt-cont-svc
/NetScaler 9000 system /barandpage.htm
bind lb vserver vsvr -policyName policy10
set lb vserver vsvr -sc ON
save config
<!--NeedCopy-->
```

Example 5 - SureConnect with content switching

This example illustrates how to configure SureConnect where the NetScaler content switching and load balancing features are being used. SureConnect is configured on a load balancing virtual server bound to a content switching virtual server.

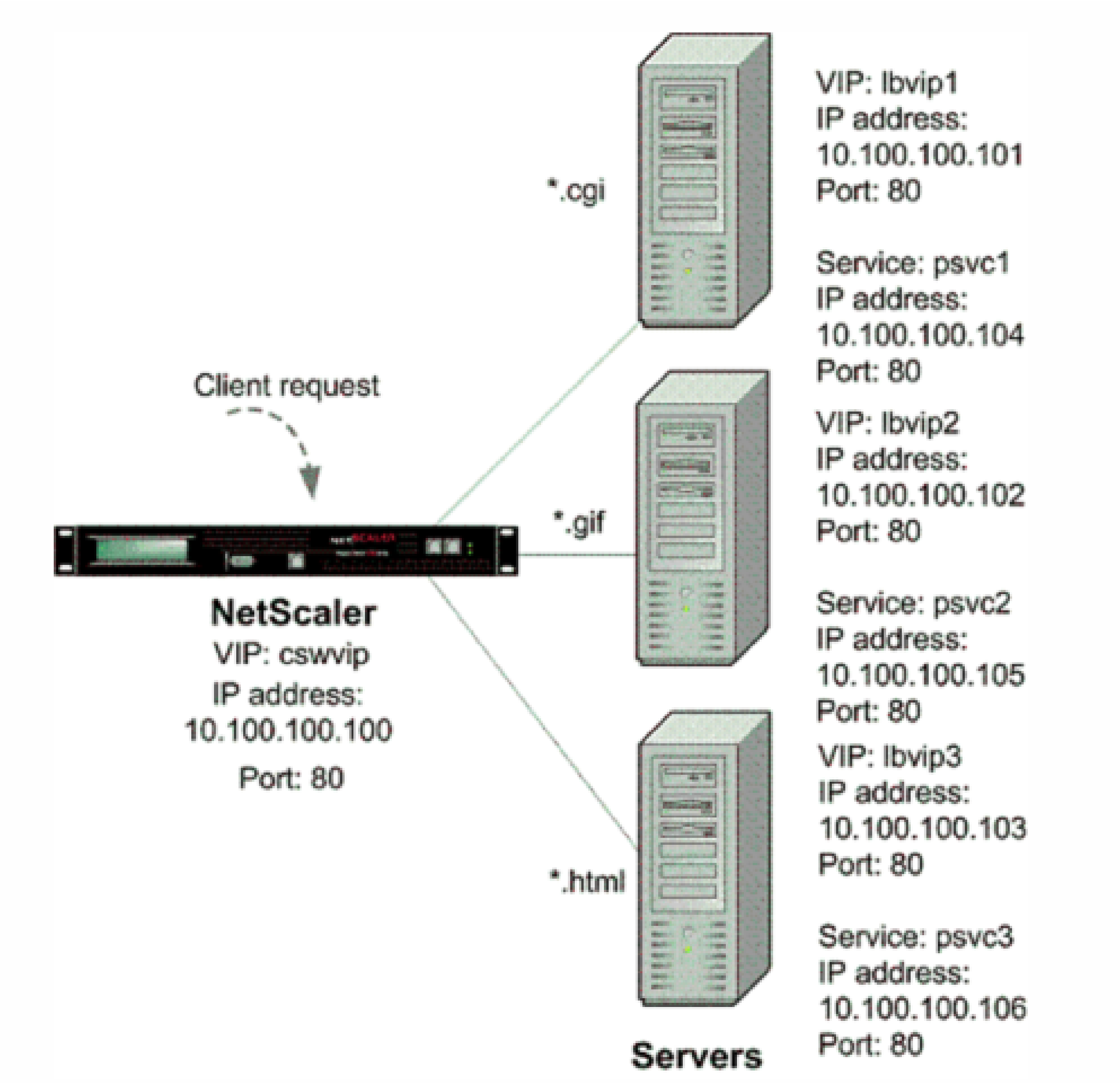
The alternate content is distributed under the content switching virtual server according to the content switching rules. For more information about load balancing and content switching, see "Load Balancing" and "Content Switching."

In this case, three physical services with IP addresses 10.100.100.104, 10.100.100.105, and 10.100.100.106 are bound to three load balancing virtual servers with IP addresses 10.100.100.101, 10.100.100.102, and 10.100.100.103. These three load balancing virtual servers are bound to a content switching virtual server with IP address 10.100.100.100.

In this setup, lbvip1 contains .cgi content, lbvip2 contains .png content, and .lbvip3 contains .html content.

The name and location of the alternate page file is specified in the file alternatepage.htm, which resides on lbvip3. The embedded objects in this file must be distributed according to the content switching rules (any embedded gif will reside on lbvip2, any embedded htm will reside on lbvip3, and so on).

Figure 3. SureConnect Configuration - Example 5


Client request
NetScaler
VIP: cswvip
IP address:
10.100.100.100
Port: 80
*.cgi
*.gif
*.html
VIP: lbvip1
IP address:
10.100.100.101
Port: 80
Service: psvc1
IP address:
10.100.100.104
Port: 80
VIP: lbvip2
IP address:
10.100.100.102
Port: 80
Service: psvc2
IP address:
10.100.100.105
Port: 80
VIP: lbvip3
IP address:
10.100.100.103
Port: 80
Service: psvc3
IP address:
10.100.100.106
Port: 80
Servers

At the NetScaler command prompt, type the following commands:

```
enable feature CS LB SC
add vserver cswvip HTTP 10.100.100.100 80 -type CONTENT
add vserver lbvip1 HTTP 10.100.100.101 80 -type ADDRESS
add vserver lbvip2 HTTP 10.100.100.102 80 -type ADDRESS
add vserver lbvip3 HTTP 10.100.100.103 80 -type ADDRESS
add service psvc1 10.100.100.104 HTTP 80
add service psvc2 10.100.100.105 HTTP 80
add service psvc3 10.100.100.106 HTTP 80
bind lb vserver lbvip1 psvc1
bind lb vserver lbvip2 psvc2
bind lb vserver lbvip3 psvc3
add cs policy CSWpolicy1 -url /*.cgi
```

```
bind cs vserver cswvip lbvip1 -policyName CSWpolicy1
add cs policy CSWpolicy2 -url /*.png
bind cs vserver cswvip lbvip2 -policyName CSWpolicy2
add cs policy CSWpolicy3 -url /*.htm
bind cs vserver cswvip lbvip3 -policyName CSWpolicy3
add sc policy SCpol -url /cgi-bin/delay.cgi -delay 4000000 -action ACS cswvip /alternatepage.htm
bind lb vserver lbvip1 -policyName SCpol
set lb vserver lbvip1 -sc ON
save config
<!--NeedCopy-->
```

Surge protection

September 21, 2020

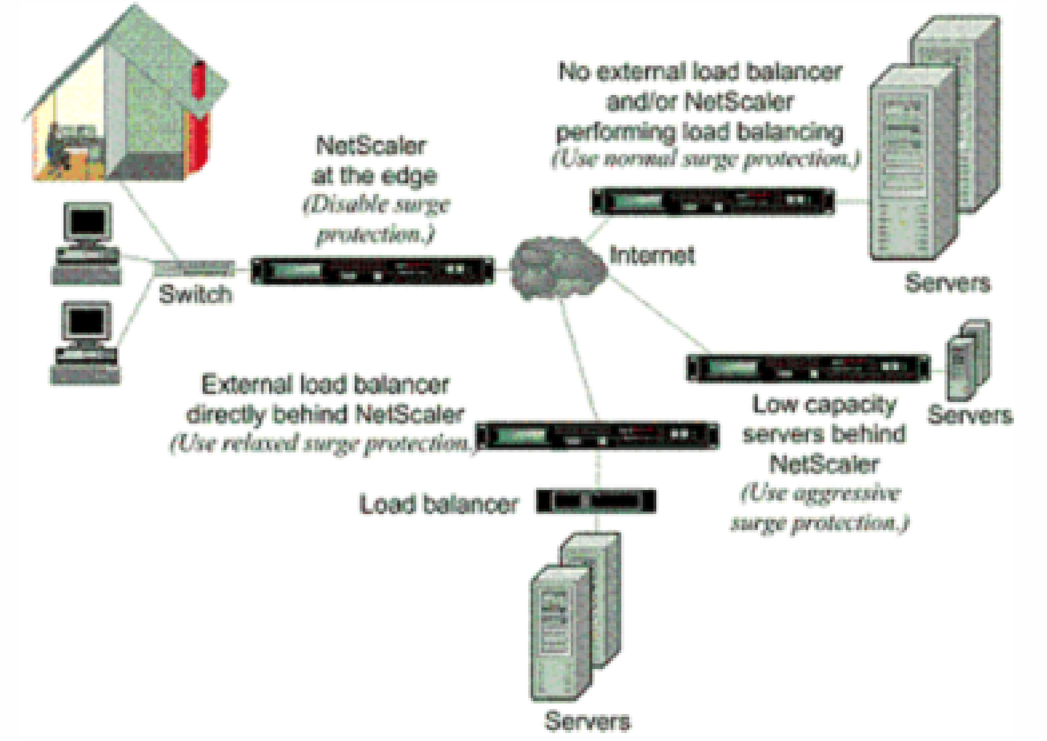
When a surge in client requests overloads a server, server response becomes slow, and the server is unable to respond to new requests. The Surge Protection feature ensures that connections to the server occur at a rate that the server can handle. The response rate depends on how surge protection is configured. The NetScaler appliance also tracks the number of connections to the server, and uses that information to adjust the rate at which it opens new server connections.

Surge protection is enabled by default. If you do not want to use surge protection, as will be the case with some special configurations, you must disable it.

The default surge protection settings are sufficient for most uses, but you can configure surge protection to tune it for your needs. First, you can set the throttle value to tell it how aggressively to manage connection attempts. Second you can set the base threshold value to control the maximum number of concurrent connections that the NetScaler appliance will allow before triggering surge protection. (The default base threshold value is set by the throttle value, but after setting the throttle value you can change it to any number you want.)

The following figure illustrates how surge protection is configured to handle traffic to a Web site.

Figure 1. A Functional Illustration of NetScaler Surge Protection


No external load balancer
and/or NetScaler
performing load balancing
(Use normal surge protection.)
NetScaler
at the edge
(Disable surge
protection.)
Switch
Internet
Servers
External load balancer
directly behind NetScaler
(Use relaxed surge protection.)
Low capacity
servers behind
NetScaler
(Use aggressive
surge protection.)
Servers
Load balancer
Servers

Note: If the NetScaler appliance is installed at the edge of the network, where it interacts with network devices on the client side of the Internet, the surge protection feature must be disabled. Surge protection must also be disabled if you enable USIP (Using Source IP) mode on your appliance.

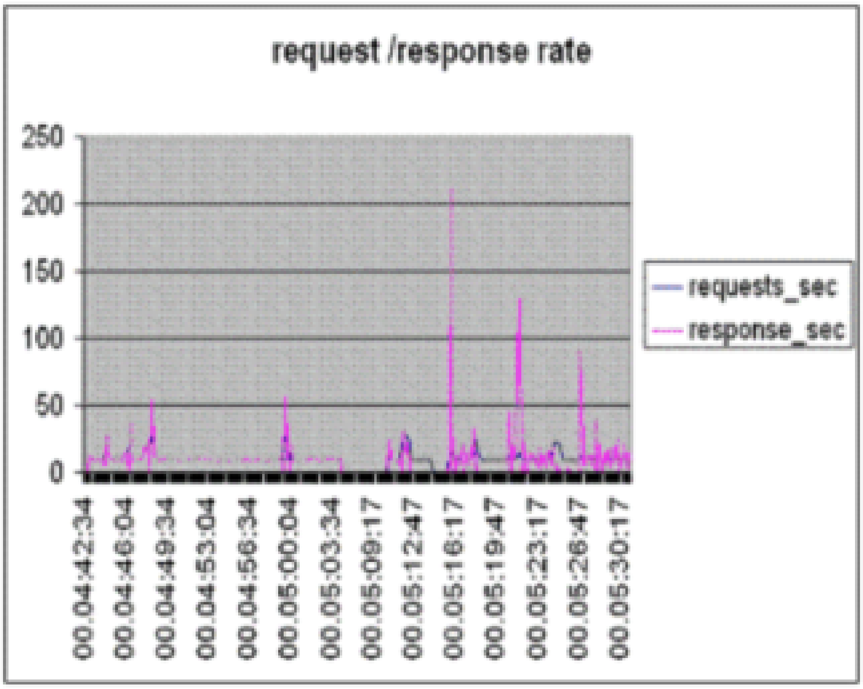
The following example and illustration show the request and response rates for two cases. In one case, surge protection is disabled, and in the other it is enabled.

When surge protection is disabled and a surge in requests occurs, the server accepts as many requests as it can process concurrently, and then begins to drop requests. As the server becomes more overloaded, it goes down and the response rate is reduced to zero. When the server recovers from the crash, usually several minutes later, it sends resets for all pending requests, which is abnormal behavior, and also responds to new requests with resets. The process repeats for each surge in requests. Therefore, a server that is under DDoS attack and receives multiple surges of requests can become unavailable to legitimate users.

When surge protection is enabled and a surge in requests occurs, surge protection manages the rate of requests to the server, sending requests to the server only as fast as the server can handle those requests. This enables the server to respond to each request correctly in the order it was received. When the surge is over, the backlogged requests are cleared as fast as the server can handle them, until the request rate matches the response rate.

The following figure compares the request and response scenarios when surge protection is enabled to that when it is disabled.

Figure 2. Request/Response Rate with and without Surge Protection



Disable and reenable surge protection

September 21, 2020

The surge protection feature is enabled by default. When surge protection is enabled, it is active for any service that you add.

Disable or reenable surge protection by using the CLI

At the command prompt, type one of the following sets of commands to disable or reenable surge protection and verify the configuration:

```
disable ns feature SurgeProtection
show ns feature
enable ns feature SurgeProtection
show ns feature
<!--NeedCopy-->
```

Example:

```
disable ns feature SurgeProtection
 Done show ns feature

```

```
        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   OFF
 .
 .
 .
 23)    HTML Injection                 HTMLInjection        ON
 24)    NetScaler Push                 push                 OFF
 Done
<!--NeedCopy-->
```

```
 enable ns feature SurgeProtection
 Done

show ns feature

        Feature                        Acronym              Status
        -------                        -------              ------
 1)     Web Logging                    WL                   ON
 2)     Surge Protection               SP                   ON
 .
 .
 .

 23)    HTML Injection                 HTMLInjection        ON
 24)    NetScaler Push                 push                 OFF
 Done

<!--NeedCopy-->
```

Disable or reenable surge protection by using the GUI

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Change Advanced Features.
3. In the Configure Advanced Features dialog box, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
4. Click OK.
5. In the Enable/Disable Feature(s) dialog box, click Yes. A message appears in the status bar, stating that the feature has been enabled or disabled.

Disable or reenable surge protection for a particular service by using the GUI

1. Navigate to Traffic Management > Load Balancing > Services. The list of configured services is displayed in the details pane.
2. In the details pane, select the service for which you want to disable or reenable the surge protection feature, and then click Open.
3. In the Configure Service dialog box, click the Advanced tab and scroll down.
4. In the Others frame, clear the selection from the Surge Protection check box to disable the surge protection feature, or select the check box to enable the feature.
5. Click OK. A message appears in the status bar, stating that the feature has been enabled or disabled.
   Note: Surge protection works only when both the feature and the service setting are enabled.

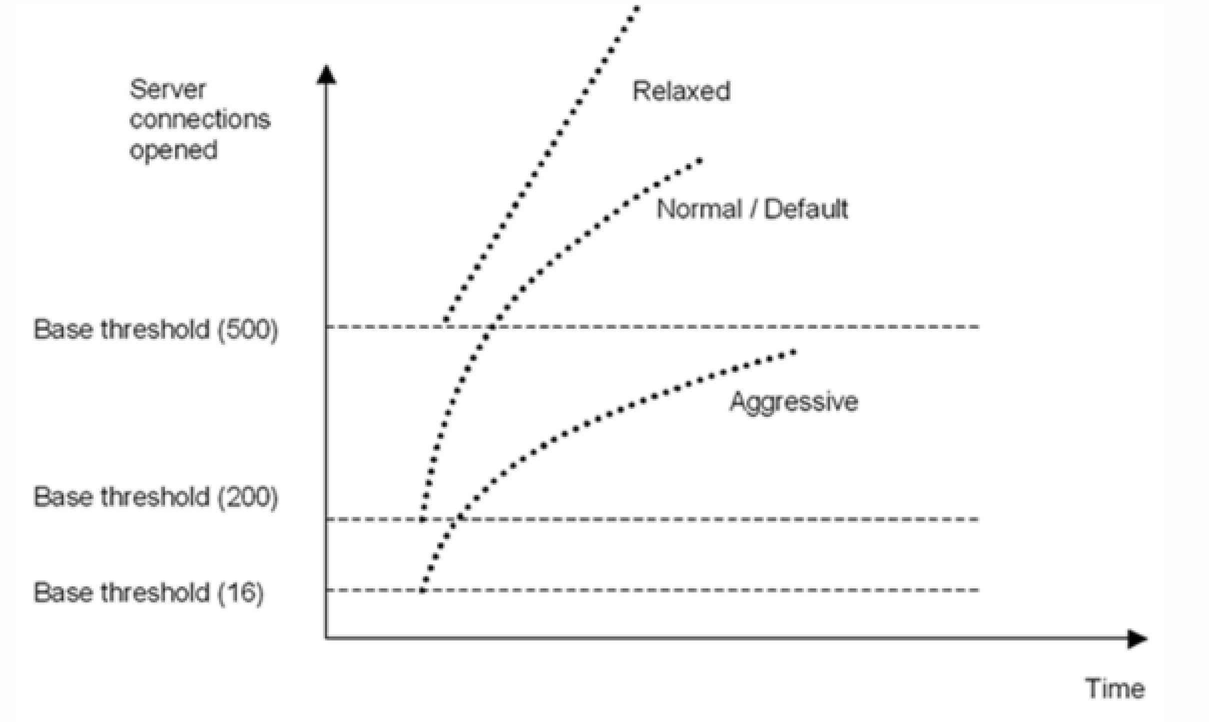
Set thresholds for surge protection

September 21, 2020

To set the rate at which the NetScaler appliance opens connections to the server, you must configure the threshold and throttle values for surge protection.

The following figure shows the surge protection curves that result from setting the throttle rate to relaxed, normal, or aggressive. Depending on the configuration of the server capacity, you can set base threshold values to generate appropriate surge protection curves.

Figure 1. Surge Protection Curves

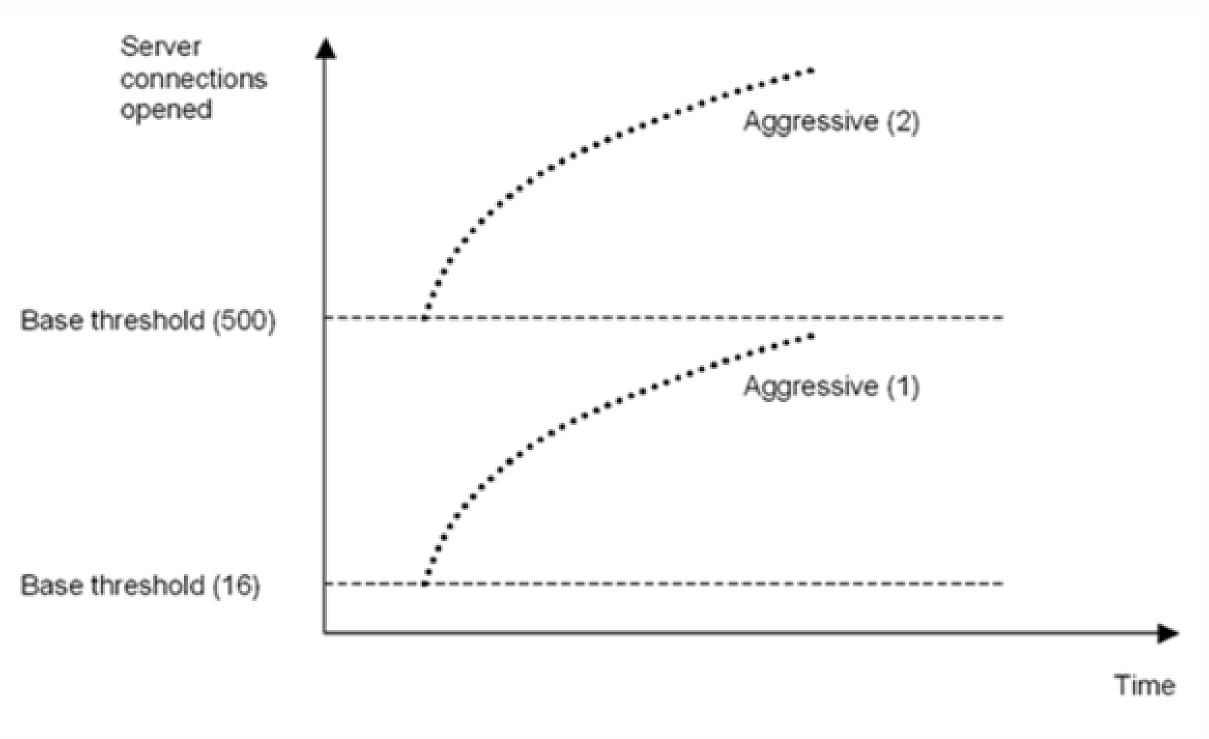


Your configuration settings affect the behavior of surge protection in the following manner:

- If you do not specify a throttle rate, it is set to normal (the default value), and the base threshold is set to 200, as shown in the preceding figure.
- If you specify a throttle rate (aggressive, normal, or relaxed) without specifying a base threshold, the curve reflects the default values of the base threshold for that throttle rate. For example, if you set the throttle rate to relaxed, the resulting curve will have the base threshold value of 500.
- If you specify only the base threshold, the entire surge protection curve shifts up or down, depending on the value you specify, as shown in the figure that follows.
- If you specify both a base threshold and a throttle rate, the resulting surge protection curve is based on the set throttle rate and adjusted according to the value set for the base threshold.

In the following figure, the lower curve (Aggressive 1) results when the throttle rate is set to aggressive but the base threshold is not set. The upper curve (Aggressive 2) results when the base threshold is set to 500, but the throttle rate is not set. The second upper curve (Aggressive 2) also results when the base threshold is set to 500, and the throttle rate is set to aggressive.

Figure 2. Aggressive Rate with the Default or a Set Base Threshold



Set the threshold for surge protection by using the GUI

1. In the navigation pane, expand System, and then select Settings.
2. In the details pane, click Global System Settings.
3. If you want to set a base threshold different from the default for the throttle rate, in the Configure Global Settings dialog box, Base Threshold text box, enter the maximum number of concurrent server connections allowed before surge protection is triggered. The base threshold is the maximum number of server connections that can be open before surge protection is activated. The maximum value for this setting is 32,767 server connections. The default setting for this value is controlled by the throttle rate you choose in the next step.
   Note: If you do not set an explicit value here, the default value will be used.
4. In the Throttle drop-down list, select a throttle rate. The throttle is the rate at which the NetScaler appliance allows connections to the server to be opened. The throttle can be set to the following values:
   - **Aggressive:** Choose this option when the connection-handling and surge-handling capacity of the server is low and the connection needs to be managed carefully. When you set the throttle to aggressive, the base threshold is set to a default value of 16, which means that surge protection is triggered whenever there are 17 or more concurrent connections to the server.

  - **Normal:** Choose this option when there is no external load balancer behind the NetScaler appliance or downstream. The base threshold is set to a value of 200, which means that surge protection is triggered whenever there are 201 or more concurrent connections to the server. Normal is the default throttle option.
  - **Relaxed:** Choose this option when the NetScaler appliance is performing load balancing between a large number of Web servers, and can therefore handle a high number of concurrent connections. The base threshold is set to a value of 500, which means that surge protection is triggered only when there are 501 or more concurrent connections to the server.

5. Click OK. A message appears in the status bar, stating that the global settings are configured.

Flush the surge queue

September 21, 2020

When a physical server receives a surge of requests, it becomes slow to respond to the clients that are currently connected to it, which leaves users dissatisfied and disgruntled. Often, the overload also causes clients to receive error pages. To avoid such overloads, the NetScaler appliance provides features such as surge protection, which controls the rate at which new connections to a service can be established.

The appliance does connection multiplexing between clients and physical servers. When it receives a client request to access a service on a server, the appliance looks for an already established connection to the server that is free. If it finds a free connection, it uses that connection to establish a virtual link between the client and the server. If it does not find an existing free connection, the appliance establishes a new connection with the server, and establishes a virtual link between client and the server. However, if the appliance cannot establish a new connection with the server, it sends the client request to a surge queue. If all the physical servers bound to the load balancing or content switching virtual server reach the upper limit on client connections (max client value, surge protection threshold or maximum capacity of the service), the appliance cannot establish a connection with any server. The surge protection feature uses the surge queue to regulate the speed at which connections are opened with the physical servers. The appliance maintains a different surge queue for each service bound to the virtual server.

The length of a surge queue increases whenever a request comes for which the appliance cannot establish a connection, and the length decreases whenever a request in the queue gets sent to the server or a request gets timed out and is removed from the queue.

If the surge queue for a service or service group becomes too long, you may want to flush it. You can flush the surge queue of a specific service or service group, or of all the services and service groups bound to a load balancing virtual server. Flushing a surge queue does not affect the existing connections. Only the requests present in the surge queue get deleted. For those requests, the client has to make a fresh request.

You can also flush the surge queue of a content switching virtual server. If a content switching virtual server forwards some requests to a particular load balancing virtual server, and the load balancing virtual server also receives some other requests, when you flush the surge queue of the content switching virtual server, only the requests received from this content switching virtual server are flushed; the other requests in the surge queue of the load balancing virtual server are not flushed.

Note:

> - You cannot flush the surge queues of cache redirection, authentication, VPN or GSLB virtual servers or GSLB services.
> - Do not use the Surge Protection feature if Use Source IP (USIP) is enabled.

Flush a surge queue by using the CLI

The flush ns surgeQ command works in the following manner:

- You can specify the name of a service, service group, or virtual server whose surge queue has to be flushed.
- If you specify a name while executing the command, surge queue of the specified entity will be flushed. If more than one entity has the same name, the appliance flushes surge queues of all those entities.
- If you specify the name of a service group, and a server name and port while executing the command, the appliance flushes the surge queue of only the specified service group member.
- You cannot directly specify a service group member `<serverName>` `and` `<port>` without specifying the name of the service group `<name>` and you cannot specify `<port>` without a `<serverName>`. Specify the `<serverName>` and `<port>` if you want to flush the surge queue for a specific service group member.
- If you execute the command without specifying any names, the appliance flushes the surge queues of all the entities present on the appliance.
- If a service group member is identified with a server name, you must specify the server name in this command; you cannot specify its IP address.

At the command prompt, type:

```
flush ns surgeQ [-name <name>] [-serverName <serverName> <port>]
<!--NeedCopy-->
```

Examples

```
1.  flush ns surgeQ  -name SVC1ANZGB  -serverName 10.10.10.1 80
The above command flushes the surge queue of the service or virtual
   server that is named SVC1ANZGB and has IP address as 10.10.10
2. flush ns surgeQ
The above command flushes all the surge queues on the appliance.
<!--NeedCopy-->
```

Flush a surge queue by using the GUI

Navigate to Traffic Management > Content Switching > Virtual Servers, select a virtual server and, in the Action list, select Flush Surge Queue.

DNS security options

September 21, 2020

You can now configure the DNS security options from the Add DNS Security Profile page in the NetScaler GUI. To configure the DNS security options from the NetScaler CLI or the NITRO API, use the AppExpert components. For instructions, see the NITRO API documentation and the NetScaler Command Reference Guide.

One option, cache poisoning protection, is enabled by default and cannot be disabled. You can apply the other options to all DNS endpoints or to specific DNS virtual servers in your deployment, as shown in the following table:

Security option	Can be applied to all DNS endpoints?	Can be applied to specific DNS virtual servers?
DNS DDoS protection	Yes	Yes
Manage exceptions – whitelist/blacklist servers	Yes	Yes
Prevent random subdomain attacks	Yes	Yes
Bypass the cache	Yes	No
Enforce DNS transactions over TCP	Yes	Yes
Provide root details in the DNS response	Yes	No

Cache poisoning protection

A cache poisoning attack redirects users from legitimate sites to malicious websites.

For example, the attacker replaces a genuine IP address in the DNS cache with a fake IP address that they control. When the server responds to requests from these addresses, the cache is poisoned and the subsequent requests for the addresses of the domain are redirected to the attacker's site.

The Cache Poisoning Protection option prevents insertion of corrupt data into the database that caches DNS server requests and responses. This is an inbuilt feature of the NetScaler appliances and is always enabled.

DNS DDoS protection

You can configure the DNS DDoS Protection option for each type of request that you suspect might be used in a DDoS attack. For each type, the NetScaler appliance drops any requests received after a threshold value for the number of requests received in a specified period of time (time slice) is exceeded. You can also configure this option to log a warning to the SYSLOG server. For example:

- **DROP:** - Assume that you have enabled A record protection with threshold value 15, time slice as 1 second and chosen DROP. When the incoming requests exceed 15 in 1 second, then the packets start getting dropped.
- **WARN:** - Assume that you have enabled A record protection with threshold value 15, time slice as 1 second and chosen WARN. When the incoming requests exceeds around 15 queries in 1 second, a warning message is logged indicating a threat and then the packets start getting dropped. It is recommended to set threshold values for WARN smaller than that of DROP for a record type. This will help administrators identify an attack by logging a warning message before the actual attack happens and NetScaler starts dropping incoming requests.

Set a threshold for incoming traffic by using the NetScaler GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profile** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
4. Expand **DNS DDoS Protection**.
    a) Select the record type and enter the threshold limit and the time slice value.
    b) Select **DROP** or **WARN**.
    c) Repeat steps a and b for each of the other record types that you want to protect against.
5. Click **Submit**.

Manage exceptions – whitelist/blacklist servers

Manage excepts enables you to add exceptions either to blacklist or whitelist domain name and IP addresses. For example:

- When a particular IP address is identified posting an attack, such IP address can be blacklisted.
- When administrators find that there is unexpectedly high request for a particular domain name, then that domain name can be blacklisted.
- NXDomains and some of the existent domains which can consume the server resources can be blacklisted.
- When administrators whitelist domain names or IP addresses, queries or requests only from these domains or IP addresses are answered and all others are dropped.

Create a whitelist or blacklist by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
   a) Expand **Manage exceptions – Whitelist/Blacklist Servers**.
   b) Select **Block** to block queries from blacklisted domains/addresses, or select **Allow** only to allow queries from whitelisted domains/addresses.
   c) In the **Domain name / IP Address** box, enter the domain names, IP addresses, or IP address ranges. Use commas to separate the entries.
   Note: If you select **Advanced Option**, you can use the "start with," "contains," and "ends with" options to set the criteria.
   For example, you can set criteria to block a DNS query that starts with "image" or ends with ".co.ru" or contains "mobilesites."
4. Click **Submit**.

Prevent random subdomain attacks

In random subdomain attacks, queries are sent to random, nonexistent subdomains of legitimate domains. This increases the load on the DNS resolvers and servers. As a result, they can become overloaded and slow down.

The Prevent Random Subdomain Attacks option directs the DNS responder to drop DNS queries that exceed a specified length.

Assume that example.com is a domain name owned by you and hence the resolution request comes to your DNS server. The attacker can append a random subdomain to example.com and send a request. Based on the specified query length and the FQDN, the random queries will be dropped.

For example, if the query is www.image987trending.example.com, it will be dropped if the query length is set to 20.

Specify a DNS query length by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
    a) Expand **Prevent Random Subdomain Attacks**.
    b) Enter the numerical value for the query length.
4. Click **Submit**.

Bypassing the cache

During an attack, the data that is already cached must be protected. To protect the cache, new requests for certain domains or record types or response codes can be sent to the origin servers instead of cached.

The Bypassing the cache option directs the NetScaler appliance to bypass the cache for specified domains, record types, or response codes when an attack is detected.

Bypass the cache for specified domains or record types or response types by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, expand **Bypassing the cache** and enter the domain names and / or choose the record types or the response types for which the cache has to be bypassed.
    - Click **Domains** and enter the domain names. Use commas to separate the entries.
    - Click **Record Types** and choose the record types.
    - Click **Response Types** and choose the response type.
4. Click **Submit**.

Enforce DNS transactions over TCP

Some DNS attacks can be prevented if the transactions are forced to use TCP instead of UDP. For example, during a bot attack, the client sends a flood of queries but cannot handle responses. If the use of TCP is enforced for these transactions, then the bots cannot understand the responses and therefore cannot send requests over TCP.

Force domains or record types to operate at the TCP level by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, expand **Enforce DNS Transactions over TCP** and enter the domain names and / or choose the record types for which the DNS transactions must be enforced over TCP.
    - Click **Domains** and enter the domain names. Use commas to separate the entries.
    - Click **Record Types** and choose the record types.
4. Click **Submit**.

Provide root details in the DNS response

In some attacks, the attacker sends a flood of queries for unrelated domains that are not configured or cached on the NetScaler appliance. lf the dnsRootReferral parameter is ENABLED, it exposes all the root servers.

The Provide Root Details in the DNS Response option directs the NetScaler appliance to restrict access to root referrals for a query that is not configured or cached. The appliance sends a blank response.

The Provide Root Details in the DNS Response option can also mitigate or block amplification attacks. When the dnsRootReferral parameter is DISABLED, there will be no root referral in the NetScaler responses and hence they do not get amplified.

Enable or disable access to the root server by using the GUI

1. Navigate to **Configuration** > **Security** > **DNS Security**.
2. On the **DNS Security Profiles** page, click **Add**.
3. On the **Add DNS Security Profile** page, do the following:
    a) Expand **Provide Root Details in the DNS Response**.
    b) Click **ON** or **OFF** to allow or restrict access to the root server.
4. Click **Submit**.

System

September 21, 2020

This section provides system-level information of the NetScaler. This includes a detailed explanation of system-level features, the scenarios in which the features can be used, the configuration steps, and examples to help you better understand the features.

- Basic Operations
- Authentication and Authorization
- TCP Configurations
- HTTP Configurations
- SNMP
- Audit Logging
- Web Server Logging
- Call Home
- Reporting Tool
- CloudBridge Connector
- High Availability
- TCP Optimization

Basic operations

September 21, 2020

Any changes you make to the configuration of a NetScaler appliance are not saved automatically. You have to save the settings manually. When an appliance is restarted, it loads the latest saved configuration.

View and save configurations

Configurations are stored in the /nsconfig/ns.conf directory. For configurations to be available across sessions, you must save the configuration after every configuration change.

To view the running configuration by using the command line interface

At the command prompt, type:

```
show ns runningConfig
```

To view the running configuration by using the GUI

- Navigate to **System** > **Diagnostics** and, in the **View Configuration group**, click **Running Configuration.**

To find the difference between two configuration files by using the command line interface

At the command prompt, type:

```
diff ns config <configfile> <configfile2>
```

To find the difference between two configuration files by using the GUI

- Navigate to **System** > **Diagnostics** and, in the **View Configuration group**, click **Configuration difference**.

To save configurations by using the command line interface

At the command prompt, type:

```
save ns config
```

To save configurations by using the GUI

On the **Configuration** tab, in the top-right corner, click the **Save** icon.

To view the saved configurations by using the command line interface

At the command prompt, type:

```
show ns ns.conf
```

To view the saved configurations by using the GUI

Navigate to **System** > **Diagnostics ** and, in the **View Configuration** group, click **Saved Configuration**.

Clear configuration

You have the following three options for clearing the NetScaler configuration.

Basic level. Clearing your configuration at the basic level clears all settings except the following:

- `NSIP, MIP(s), and SNIP(s)`
- `Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)`
- `HA node definitions`
- `Feature and mode settings`
- `Default administrator password (nsroot)`

Extended level. Clearing your configuration at the extended level clears all settings except the following:

- `NSIP, MIP(s), and SNIP(s)`
- `Network settings (Default Gateway, VLAN, RHI, NTP, and DNS settings)`
- `HA node definitions`

Feature and mode settings revert to their default values.

Full level. Clearing your configuration at the full level returns all settings to their factory default values. However, the NSIP and default gateway are not changed, because changing them could cause the appliance to lose network connectivity.

To clear the configuration by using the command line interface

At the command prompt, type:

`clear ns config -force`

Example: To forcefully clear the basic configurations on an appliance.

```
clear ns config -force basic
<!--NeedCopy-->
```

To clear the configuration by using the GUI

- Navigate to **System** > **Diagnostics** and, in the Maintenance group, click **Clear Configuration** and select the configuration level to be cleared from the appliance.

Clock synchronization

September 21, 2020

You can configure your NetScaler appliance to synchronize its local clock with a Network Time Protocol (NTP) server. This ensures that its clock has the same date and time settings as the other servers on your network.

You can configure clock synchronization on your appliance by adding NTP server entries to the ntp.conf file from either the GUI or the command line interface, or by manually modifying the ntp.conf file and then starting the NTP daemon (NTPD). The clock synchronization configuration does not change if the appliance is restarted, upgraded, or downgraded. However, the configuration does not get propagated to the secondary NetScaler in a high availability setup.

> **Note**:
> If you do not have a local NTP server, you can find a list of public, open access, NTP servers at the official NTP site,

http://www.ntp.org, under Public Time Servers List. Before configuring your NetScaler to use a public NTP server, be sure to read the Rules of Engagement page (link included on all Public Time Servers pages).

In NetScaler release 11, the NTP version has been updated from 4.2.6p3 to 4.2.8p2.

Set clock synchronization

To configure clock synchronization, you must add NTP servers and then enable NTP synchronization.

To add an NTP server by using the command line interface

At the command prompt, type the following commands to add an NTP server and verify the configuration:

- `add ntp server (<serverIP> | <serverName>)[-minpoll <positive_integer>] [-maxpoll <positive_integer>]`
- `show ntp server`

Example

```
> add ntp server 10.102.29.30 -minpoll 6 -maxpoll 11
```

To configure an NTP server by using the GUI

Navigate to **System** > **NTP Servers**, and create the NTP server.

Start NTP daemon

When you enable NTP synchronization, the NetScaler starts the NTP daemon and uses the NTP server entries in the ntp.conf file to synchronize its local time setting. If you do not want to synchronize the appliance time with the other servers in the network, you can disable NTP synchronization, which stops the NTP daemon (NTPD).

To enable NTP synchronization by using the command line interface

At the command prompt, type one of the following commands:

```
enable ntp sync
```

To enable NTP synchronization by using the GUI

Navigate to **System** > **NTP Servers**, click **Action** and select NTP Synchronization.

Configure clock synchronization

You can configure clock synchronization manually by logging on to the NetScaler appliance and editing the ntp.conf file.

To enable clock synchronization on your NetScaler appliance by modifying the ntp.conf file

1. Log on to the command line interface.
2. Switch to the shell prompt.

3. Copy the `/etc/ntp.conf` file to `/nsconfig/ntp.conf`, unless the `/nsconfig directory` already contains an `ntp.conf` file.
4. For each NTP server you want to add, you must add the following two lines to the `/nsconfig/ntp.conf` file:

```
server <IP address for NTP server> iburst

restrict <IP address for NTP server> mask <netmask> nomodify notrap nopeer noquery
```

> **Note**:
> For security reasons, there should be a corresponding restrict entry for each server entry.

Example:

In the following example, an administrator has inserted # characters to "comment out" an existing NTP entry, and then added a new entry:

```
##server 1.2.3.4 iburst

##restrict 1.2.3.4 mask 55.255.255.255 nomodify notrap nopeer noquery

server 10.102.29.160 iburst

restrict 10.102.29.160 mask 255.255.255.255 nomodify notrap nopeer noquery
```

5. If the /nsconfig directory does not contain a file named rc.netscaler, create the file.

1. Add the following entry to `/nsconfig/rc.netscaler`: `/bin/sh /etc/ntpd_ctl full_start`

This entry starts the ntpd service, checks the ntp.conf file, and logs messages in the /var/log directory.

This process runs every time the NetScaler is restarted.

7. Restart the NetScaler appliance to enable clock synchronization. Or, to start the time synchronization process without restarting the appliance, enter the following commands at the shell prompt:

- `rm /etc/ntp.conf`
- `ln -s /nsconfig/ntp.conf /etc/ntp.conf`
- `/bin/sh /etc/ntpd_ctl full_start`

Session timeout

September 21, 2020

A session timeout interval is provided to restrict the time duration for which a session (GUI, CLI, or API) remains active when not in use. For the NetScaler, the system session timeout can be configured at the following levels:

- **User level timeout.** Applicable to the specific user.

Interface type	Time out configuration
GUI	Navigate to System > User Administration > Users, select a user, and edit the user's timeout setting.
CLI	At the command prompt, enter the following command: `set system user <name> -timeout <secs>`|

- **User group level timeout.** Applicable to all users in the

Interface type	Time out configuration
GUI	Navigate to System > User Administration > Groups, select a group, and edit the group's timeout setting.
CLI	At the command prompt, enter the following command:`set system group <groupName > -timeout <secs>`|

- **Global system timeout.** Applicable to all users and users from groups who do not have a timeout configured.

Interface type	Time out configuration
GUI	Navigate to System > Settings, click Change global system settings, and update the timeout value as required.
CLI	At the command prompt, enter the following command: `set system parameter -timeout <secs>`

The timeout value specified for a user has the highest priority. If timeout is not configured for the user, the timeout configured for a member group is considered. If timeout is not specified for a group (or the user does not belong to a group), the globally configured timeout value is considered. If timeout is not configured at any level, the default value of 900 seconds is set as the system session timeout.

Additionally, you can specify timeout durations for each of the interfaces you are accessing. However, the timeout value specified for a specific interface is restricted to the timeout value configured for the user that is accessing the interface. For example, let us consider an user ”publicadmin” who has a timeout value of 20 minutes. Now, when accessing an interface, the user must specify a timeout value that is within 20 minutes.

> **Note:**
> You can choose to keep a check on the minimum and maximum timeout values by specifying the timeout as restricted (in CLI by specifying the *restrictedTimeout* parameter). This parameter is provided to account for previous NetScaler versions where the timeout value was not restricted.

- When enabled, the minimum configurable timeout value is 5 minutes (300 secs) and the maximum value is 1 day (86400 secs). If the timeout value is already configured to a value larger than 1 day, when this parameter is enabled, you are prompted to change it. If you do not change the value, the timeout value will automatically be reconfigured to the default timeout duration of 15 minutes (900 secs) on next reboot. The same will happen is the configured timeout value is less than 5 minutes.
- When disabled, the configured timeout durations are considered.

To configure the timeout duration at each interface:

Interface type	Time out configuration
CLI	Specify the timeout value on the command prompt by using the following command: `set cli mode -timeout <secs >`
API	Specify the timeout value in the login payload.

System date and time

September 21, 2020

To change the system date and time, you must use the shell interface to the underlying FreeBSD OS. However, to view the system date and time, you can use the command line interface or the GUI.

To view the system date and time by using the command line interface

At the command prompt, type:

```
show ns config
```

To view the system date and time by using the GUI

Navigate to **System** and select the **System Information** tab to view the system date.

Backup and restore

September 21, 2020

When your appliance gets corrupted or needs an upgrade, you can back up your system configuration. The procedure is done either through the Citrix CLI or GUI interface. The appliance also enables you to import the backup file from an external source. However, you can do this only through the GUI interface and there is no support through the CLI interface.

Points to remember:

- Sysid and Interfaces are the same.
- Network configuration is supported on new platform.
- new platform build is either same of backup file or higher version than it.

Back up a NetScaler appliance

Depending on the type of data to be backed up and the frequency at which you create a backup, you can take a basic backup or a full backup.

- **Basic backup.** Backs up only configuration files. You might want to perform the backup type frequently, because the files it backs up change constantly. The files that are backed up are:

For more information about the basic backup details, see the Table topic.

- **Full backup.** In addition to the files that are backed up by a basic backup, a full backup backs up some less frequently updated files. The files that are backed up when using the full backup option are:

Directory	Sub-Directory or Files
`nsconfig`	ssl, *license*, fips*
/var/	netscaler/ssl/, *wi/-java_home/jre/lib/security/cacerts/*, wi/-java_home/lib/security/cacerts/*

The backup is stored as a compressed TAR file in the `/var/ns_sys_backup/` directory. To avoid issues due to non-availability of disk space, you can store a maximum of 50 backup files in this directory. You can use the rm system backup command to delete existing backup files so that you can create more backups.

> **Note:**
>
> While the backup operation is in progress, do not execute commands that affect the configuration.
>
> If a file that is required to be backed up is not available, the operation skips that file.

To back up a NetScaler appliance

To back up the NetScaler by using the NetScaler command line interface

At the command prompt, do the following:

1. Save the NetScaler configurations.

```
save ns config
```

1. Create the backup file.

```
create system backup [<fileName>] -level <basic | full> -comment <string>
```

> **Note:**
>
> If the file name is not specified, the appliance creates a TAR file with the following naming convention:`backup_<level>_<nsip_address>_<date-timestamp>.tgz`.

Example: To back up the full appliance using the default naming convention for the backup file.

```
> create system backup -level full
```

1. Verify that the backup file was created.

```
show system backup
```

You can view properties of a specific backup file by using the file name parameter.

To back up the appliance by using the NetScaler GUI

Navigate to **System** > **Backup and Restore**, click **Backup** and then specify the details of the backup.

Restore a NetScaler appliance

> **Important**
>
> The restore operation does not succeed if the backup file is renamed or if the contents of the file are modified.

When you restore the appliance from a backup file, the restore operation untars the backup file into the `/var/ns_sys_backup/` directory. Once the untar operation is complete, the files are copied to their respective directories.

To restore the NetScaler from a local backup file by using the command line interface

Note: Citrix recommends backing up the current configuration before restoring a previous configuration. However, if you do not want the restore command to automatically create a backup of the current configuration, use the **–skipBackup** parameter.

At the command prompt, do the following:

1. Obtain a list of the backup files available on the appliance.

   ```
   show system backup
   ```

2. Restore the appliance by specifying one of the backup files.

   ```
   restore system backup <filename> [-skipBackup]
   ```

   Example: To restore by using a full backup of an appliance

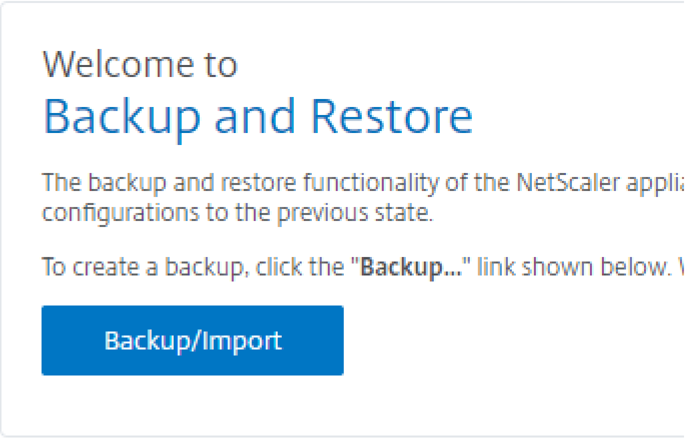
   ```
   > restore system backup backup_full_<nsip_address>_<date-timestamp>.tgz
   ```

3. Reboot the appliance.

   ```
   reboot
   ```

To backup and restore an appliance by using the NetScaler GUI

1. Navigate to **System** > **Backup and Restore**.



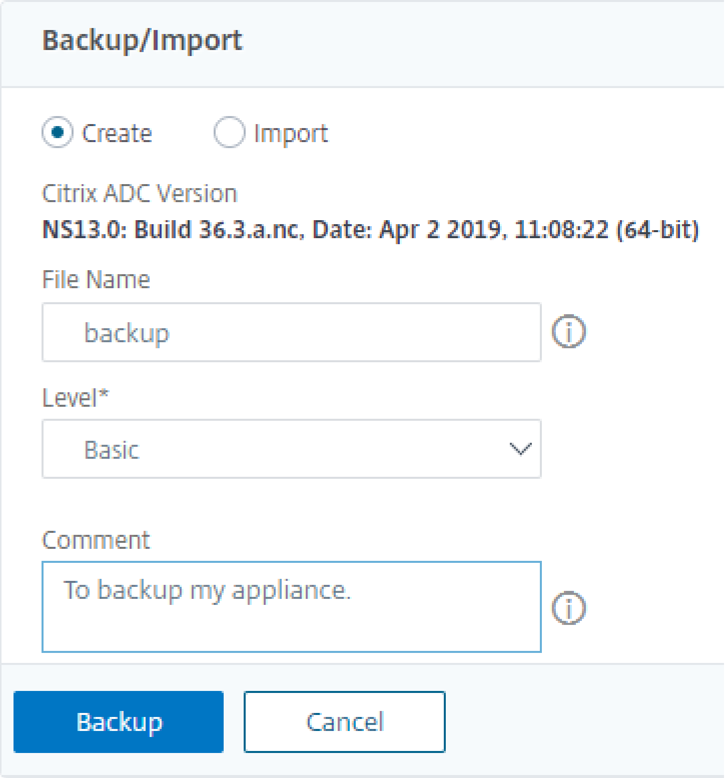
Welcome to
Backup and Restore

The backup and restore functionality of the NetScaler appli
configurations to the previous state.

To create a backup, click the "**Backup...**" link shown below.

2. Click **Backup/Import** to start the process.

3. In the **Backup/Import** page, select **Create** and set the following parameters.
    a) File name. Name of the backup file.
    b) Level. Select a backup level as basic or full.
    c) Comment. Provide a brief description for the backup.
4. Click **Backup**.



Backup/Import

● Create ○ Import

Citrix ADC Version

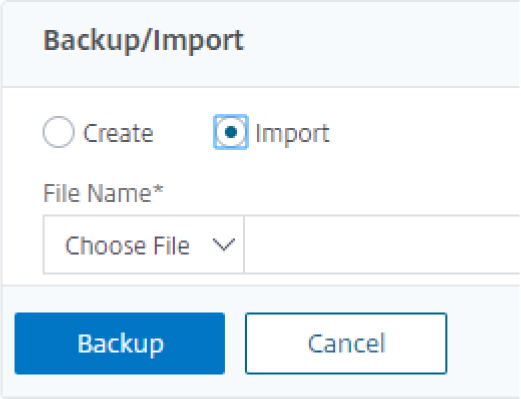
NS13.0: Build 36.3.a.nc, Date: Apr 2 2019, 11:08:22 (64-bit)

File Name: backup

Level*: Basic

Comment: To backup my appliance.

5. If you want to import a backup, you must select **Import**.



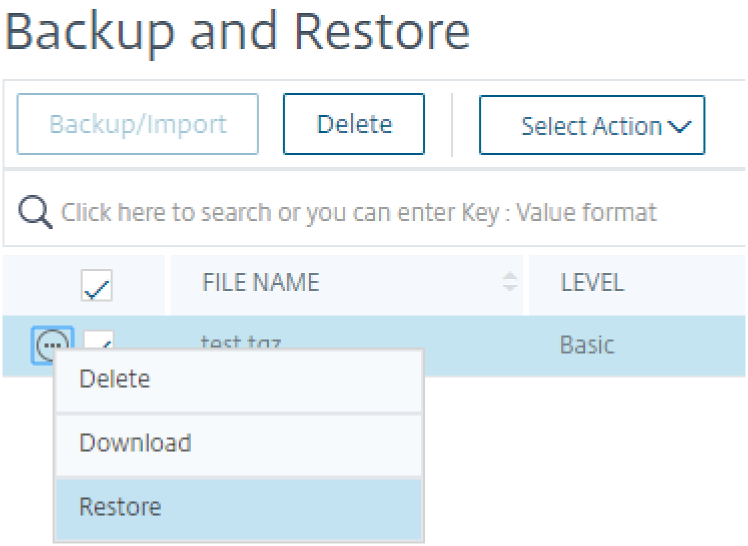
Backup/Import

○ Create ● Import

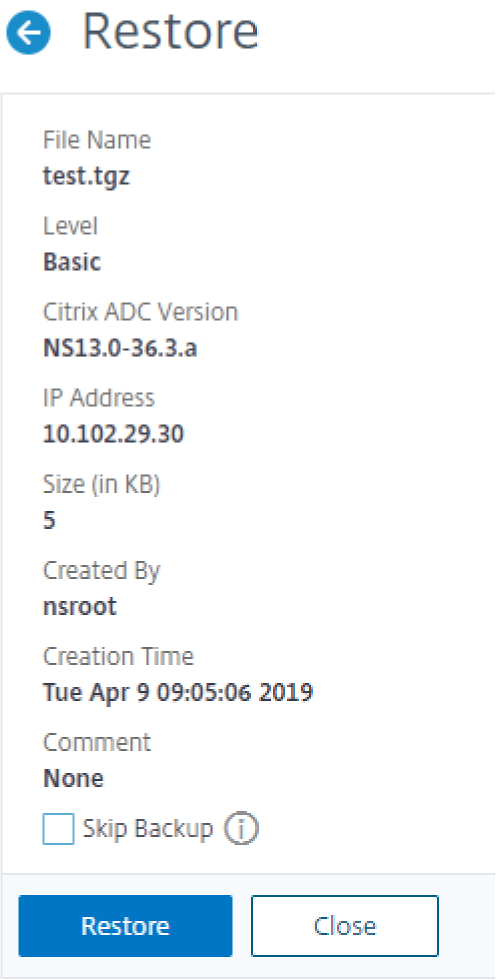
File Name*: Choose File

6. Once backup is complete, you can select the file and click **Download**.

7. To restore, select the backup file and click **Restore**.


Backup and Restore
Backup/Import
Delete
Select Action
Click here to search or you can enter Key : Value format
FILE NAME
LEVEL
test.tgz
Basic
Delete
Download
Restore

8. In the **Restore** page, verify backup file details and click **Restore**.



Restore

File Name
test.tgz

Level
Basic

Citrix ADC Version
NS13.0-36.3.a

IP Address
10.102.29.30

Size (in KB)
5

Created By
nsroot

Creation Time
Tue Apr 9 09:05:06 2019

Comment
None

☐ Skip Backup

9. After you restore, you must reboot the appliance.

For more information on how to backup and restore an SDX appliance, see Backup and restore SDX appliance

For information about operations performed on system backup, see System Backup topic.

Restart or shut down

September 21, 2020

The NetScaler appliance can be remotely restarted or shut down from the available user interfaces. When a standalone NetScaler appliance is restarted or shut down, the unsaved configurations (configurations performed since the last
save ns config command was issued) are lost.

In a high availability setup, when the primary appliance is rebooted/shut down, the secondary appliance takes over and becomes the primary. The unsaved configurations from the old primary are available on the new primary appliance.

You can also restart the appliance by only rebooting the NetScaler software and not rebooting the underlying operating system. This is called a warm reboot. For example, when you add a new license or change the NetScaler IP address, you can warm reboot the NetScaler appliance for these changes to take place.

> **Note**
>
> Warm reboot can be performed only on nCore appliances.

To restart by using the command line interface

At the command prompt, type:

reboot [-warm]

To restart the NetScaler by using the GUI

1. In the GUI, click Reboot on the home page of the Configuration tab.
2. When prompted to reboot, select Save configuration to make sure that you do not lose any configurations.

> **Note**
> You can perform a warm reboot by selecting Warm reboot.

To shut down the NetScaler by using the command line interface

At the command prompt, type:

- shutdown –p now: Shuts down the software and switches off the NetScaler. To restart NetScaler MPX, press the AC power switch. To Restart NetScaler VPX , restart the VPX instance.
- shutdown –h now: Shuts down the software and leaves the NetScaler switched on. Press any key to restart the NetScaler. This command does not switch off the NetScaler. Therefore, do not switch off the AC power or remove the AC power cables.

> **Note**
>
> The appliance cannot be shut down from the GUI.

Generate technical support bundle

September 21, 2020

For help with analyzing and resolving any issues with a NetScaler appliance, you can generate a technical support bundle on the appliance and send the bundle to Citrix technical support. The NetScaler technical support bundle is a zipped tar archive of system configuration data and statistics. It collects the following data from the NetScaler appliance on which you generate the bundle:

- **Configuration files**. All files in the /flash/nsconfig directory.
- **Newnslog files**. The currently running newnslog and some previous files. To minimize the archive file size, newnslog collection is restricted to 500 MB, 6 files, or 7 days, whichever occurs first. If older data is needed, it might require manual collection.
- **Log files**. Files in /var/log/messages, */var/log/ns.log*, and other files under /var/log and /var/nslog.
- **Application core files**. Files created in the /var/core directory within the last week, if any.
- **Output of some CLI show commands**.
- **Output of some CLI stat commands**.
- **Output of BSD shell commands**.

You can use a single command to generate the technical support bundle and securely upload it to the Citrix Technical Support server. To upload, you must specify your Citrix credentials. When you generate the bundle, you can specify the case or service request number that was allotted to you by Citrix Technical Support. If you have already generated a technical support bundle, you can upload the existing archive file to the Citrix Technical Support server by specifying the file name with the full path.

The technical support bundle is saved on the NetScaler appliance in an archive at the following location:

`/var/tmp/support/support.tgz`

The above path is a symlink to the most recent collector for easy access. The full filename varies, depending on the deployment topology, but generally follows a format similar to:

`collector_<P/S>_<NS IP>_<DateTime>.tgz.`

If your NetScaler appliance does not have direct Internet connectivity, you can use a proxy server to directly upload the technical support bundle to the Citrix Technical Support server. The basic format of the proxy string is:

`proxy_IP:<proxy_port>`

If the proxy server requires authentication, the format is:

`username:password@proxsy_IP:<proxy_port>`

> **Note**
>
> For NetScaler appliances in a high availability pair, you must generate the technical support bundle on each of the two nodes.
>
> For NetScaler appliances in a Cluster setup, you can generate the technical support bundle on

> each node individually, or you can generate smaller abbreviated archives for all nodes by using the cluster IP address.
>
> For NetScaler Admin Partitions, you must generate the technical support bundle from the default admin partition. To get the technical support bundle for a specific partition, you must specify the name of the partition for which you want to generate the technical support bundle. If you do not specify the name of the partition, data is collected from all admin partitions.

Download technical support bundle from NetScaler MPX and VPX appliances

To run a collector file by using the NetScaler GUI, you must complete the following procedure:

> Note: The procedure is applicable for software release 9.2 or later.

1. Navigate to **System** > **Diagnostic**.
2. In the Technical Support Tools section, click **Generate Support File** link.
3. In the **Tech Support** page, set the following parameters:
   a) Scope. To collect data from one or more nodes.
   b) Partition. Name of the partition.
   c) Citrix Technical Support pload Options. Set all options such as proxy server, service case number, collector archive file name. and a brief description of the archive file for uploading the technical support bundle.
   d) Citrix Account. Enter your Citrix credentials.
4. Click **Run**.
5. The Technical Support bundle is generated.
6. Click **Yes** to download the Technical support bundle to your local desktop.

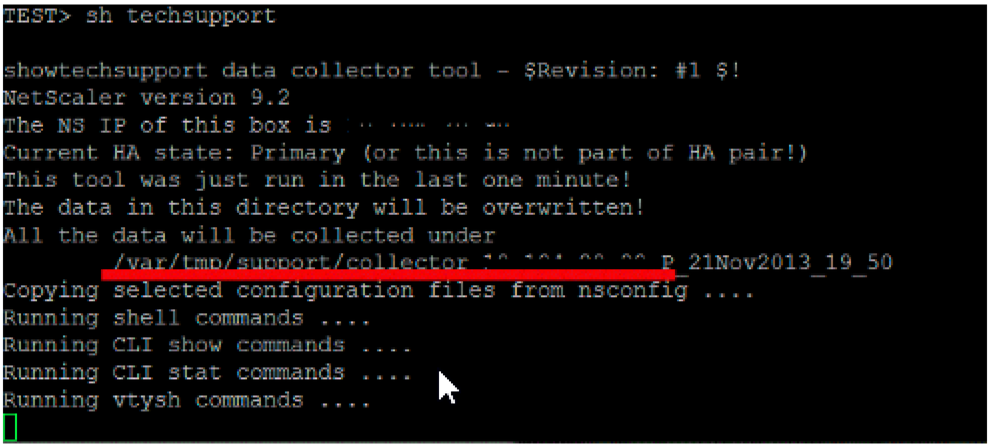
Obtain technical support bundle by using the command interface

1. Download the file from the appliance using a Secure FTP (SFTP) or Secure Copy (SCP) utility, such as WinSCP, and upload it to Citrix Insight Services for analysis.

   Note: In NetScaler software release earlier than 9.0, the collector script must be downloaded separately and executed.

   ```
   > show techsupport -scope CLUSTER
   ```

2. This collects show techsupport information from all nodes in the cluster and compress the files into a single archive.
3. After the appliance generates the collector archive the location of the file is displayed as shown in the following screenshot.



```
TEST> sh techsupport

showtechsupport data collector tool - $Revision: #1 $!
NetScaler version 9.2
The NS IP of this box is [redacted]
Current HA state: Primary (or this is not part of HA pair!)
This tool was just run in the last one minute!
The data in this directory will be overwritten!
All the data will be collected under
        /var/tmp/support/collector_[redacted]_P_21Nov2013_19_50
Copying selected configuration files from nsconfig ....
Running shell commands ....
Running CLI show commands ....
Running CLI stat commands ....
Running vtysh commands ....
```

The file is stored in `/var/tmp/support` and you can verify it by logging into a Citrix ADC appliance and running the following command from a shell prompt.

```
root@NS## cd /var/tmp/support/
root@NS## ls -l
```

Upload and run collector script on the NetScaler appliance

To upload and run the updated collector script on a NetScaler appliance, complete the following procedure:

> **Note**: The procedure is applicable for software release 9.2 or later.

1. Copy the updated collector script showtechsupport.pl (attached with this article) to the `/nsconfig` directory of the Citrix ADC appliance. Use an SFTP or SCP utility, such as WinSCP, to copy the file to the Citrix ADC appliance.
2. Run the following commands:

```
     > shell
     # cd /nsconfig
     # chmod +x showtechsupport.pl
     # ./showtechsupport.pl
<!--NeedCopy-->
```

1. Switch to the CLI of the appliance.

2. Run the following command to create log files for analysis:

   `> show techsupport`

   The `collector_<NSIP>_<P/S>_<DateTime>.tgz` file is created in the `/var/tmp/support` directory of the appliance.
   Note: You must then upload the .tgz file to Citrix Insight Services for analysis.

3. If you must uninstall this script, run the following commands from the CLI of the appliance:

   ```
   > shell
   rm /nsconfig/showtechsupport.pl
   ```

Obtain diagnostic bundle from NetScaler SDX

1. Open the NetScaler SDX Configuration Utility with the NetScaler SDX Management IP address from a web browser.
2. Expand the **Diagnostics** node.
3. Select the **Technical Support** node.
4. Click Generate Technical Support File.
5. Select **Appliance** (Including Instances) from the drop-down menu.
6. Click **Add**.
7. Select one or more instances to add in.
8. Click **OK**. Wait for the process to complete.
9. Select the bundle name that was generated and then click **Download**
10. Upload the bundle file to Citrix Insight Services.

Additional Resources

Watch a Video

Read another topic

Command Reference Doc

Authentication and authorization

September 21, 2020

To configure NetScaler authentication and authorization, you must first define the users who have access to the NetScaler appliance, and then you can organize these users into groups. After configuring users and groups, you need to configure command policies to define types of access, and assign the policies to users and/or groups.

You must log on as an administrator to configure users, groups, and command policies. The default NetScaler administrator user name is *nsroot*. After logging on as the default administrator, you should change the password for the nsroot account. Once you have changed the password, no user can access the NetScaler appliance until you create an account for that user. If you forget the administrator password after changing it from the default, you can reset it to nsroot.

> **Note**
>
> Local users can authenticate to the NetScaler even if external authentication servers are configured. You can restrict this by disabling the localAuth parameter of the set system parameter command.

Configuring users, user groups, and command policies

September 21, 2020

You must define users by configuring accounts for them. To manage user accounts, you can organize them into groups. You can create command policies, or use built-in command policies, to regulate user access to commands.

You can also customize the command-line prompt for a user. Prompts can be defined in a user configuration, user-group configuration, or global system configuration. The prompt displayed for a given user is determined by the following order of precedence:

1. Display the prompt as defined in the user's configuration.
2. Display the prompt as defined in the group configuration for the user's group.
3. Display the prompt as defined in the system global configuration settings.

You can now specify a timeout value for inactive CLI sessions for a system user. If a user session is idle for an exceeded time, the appliance terminates the connection. The timeout is defined in the user, user-group, or global system configuration settings. The timeout for inactive CLI sessions for a user is determined by the following order of precedence:

1. Time out value as defined in the user's configuration.
2. Time out value as defined in the group configuration for the user's group.
3. Time out value as defined in the global system configuration settings.

A root administrator can configure maximum concurrent session limit for a system user. By restricting the limit, you can reduce the number of open connections and improve server performance. As long as the CLI count is within the configured limit, concurrent users can log on the GUI any number of times. However, if the number of CLI sessions reaches the configured limit, users can no longer log on to the GUI. For example, if the number of concurrent sessions is configured to 20, concurrent users can log on to 19 CLI sessions. But if the user is logged on to a CLI session, any attempt to a GUI, CLI, or NITRO session causes an error.

> **Note**
>
> The default the number of concurrent sessions is configured to 20 and the maximum number of concurrent sessions is configured to 40.

Configuring user accounts

To configure user accounts, you simply specify user names and passwords. You can change passwords and remove user accounts at any time.

> **Note:**
>
> All characters in a password are not accepted. However, it works if you type the characters within quotes.
>
> Also, the string should not exceed a maximum length of 127 characters.

To create a user account by using the command line interface

At the command prompt, type the following commands to create a user account and verify the configuration:

- `add system user <username> [-externalAuth ( ENABLED | DISABLED )] [-promptString <string>] [-timeout \<secs>] [-logging ( ENABLED | DISABLED )] [-maxsession <positive_integer>]`
- `show system user <userName>`

Where Logging option is for external users to collect log data externally using weblogging or audit logging mechanism. If enabled, the auditing client authenticates itself with NetScaler to collect logs through this user account.

Example

```
> add system user johnd -promptString user-%u-at-%T
```

```
Enter password:

Confirm password:

> show system user johnd

user name: john

     Timeout:900 Timeout Inherited From: Global

     External Authentication: ENABLED

     Logging: DISABLED

     Maximum Client Sessions: 20
<!--NeedCopy-->
```

To configure a user account by using the GUI

Navigate to **System** > **User Administration** > **Users**, and create the user.

Configuring user groups

After configuring a user group, you can easily grant the same access rights to everyone in the group. To configure a group, you create the group and bind users to the group. You can bind each user account to more than one group. Binding user accounts to multiple groups might allow more flexibility when applying command policies.

To create a user group by using the command line interface

At the command prompt, type the following commands to create a user group and verify the configuration:

- `add system group <groupName> [-promptString <string>] [-timeout <secs>]`
- `show system group <groupName>`

Example

```
> add system group Managers -promptString Group-Managers-at-%h
```

To bind a user to a group by using the command line interface

At the command prompt, type the following commands to bind a user account to a group and verify the configuration:

- `bind system group <groupName> -userName <userName>`
- `show system group <groupName>`

Example

```
> bind system group Managers -userName user1
```

To configure a user group by using the GUI

Navigate to **System** > **User Administration** > **Groups**, and create the user group.

> **Note**:
>
> To add members to the group, in the Members section, click **Add**. Select users from the Available list and add them to the Configured list.

Configuring command policies

Command policies regulate which commands, command groups, virtual servers, and other entities that users and user groups are permitted to use.

The appliance provides a set of built-in command policies, and you can configure custom policies. To apply the policies, you bind them to users and/or groups.

Here are the key points to keep in mind when defining and applying command policies.

- You cannot create global command policies. Command policies must be bound directly to the users and groups on the appliance.
- Users or groups with no associated command policies are subject to the default (DENY-ALL) command policy, and are therefore unable to execute any configuration commands until the proper command policies are bound to their accounts.
- All users inherit the policies of the groups to which they belong.
- You must assign a priority to a command policy when you bind it to a user account or group account. This enables the appliance to determine which policy has priority when two or more conflicting policies apply to the same user or group.
- The following commands are available by default to any user and are unaffected by any command you specify:
- help, show CLI attribute, set CLI prompt, clear cli prompt, show cli prompt, alias, unalias, history, quit, exit,whoami, config, set cli mode, unset cli mode, and show cli mode.

Built-in Command Policies

The following table describes the built-in policies.

Table 1. Built-in Command Policies

Policy name	Allows
read-only	Read-only access to all show commands except show ns runningConfig, show ns ns.conf, and the show commands for the NetScaler command group.
operator	Read-only access and access to commands to enable and disable services and servers.
network	Full access, except to the set and unset SSL commands, show ns ns.conf, show ns runningConfig, and show gslb runningConfig commands.

Policy name	Allows
sysadmin	[Included in NetScaler 12.0 and later] A sysadmin is lower than a superuser is terms of access allowed on the appliance. A sysadmin user can perform all NetScaler operations with the following exceptions: no access to the NetScaler shell, cannot perform user configurations, cannot perform partition configurations, and some other configurations as stated in the sysadmin command policy.
superuser	Full access. Same privileges as the `nsroot` user.

Creating custom command policies

Regular expression support is offered for users with the resources to maintain more customized expressions, and for those deployments that require the flexibility that regular expressions offer. For most users, the built-in command policies are sufficient. Users who need more levels of control but are unfamiliar with regular expressions might want to use only simple expressions, such as those in the examples provided in this section, to maintain policy readability.

When you use a regular expression to create a command policy, keep the following in mind.

- When you use regular expressions to define commands that will be affected by a command policy, you must enclose the commands in double quotation marks. For example, to create a command policy that includes all commands that begin with show, type the following:
- "^show .*$"
- To create a command policy that includes all commands that begin with rm, type the following:
- "^rm .*$"
- Regular expressions used in command policies are not case sensitive.

The following table lists examples of regular expressions:

Table 2. Examples of Regular Expressions for Command Policies

Command specification	Matches these commands
“^rm\s+.*$”	All remove actions, because all remove actions begin with the rm string, followed by a space and more parameters such as command groups, command object types, and arguments.
“^show\s+.*$”	All show commands, because all show actions begin with the show string, followed by a space and more parameters such as command groups, command object types, and arguments.
“^shell$”	The shell command alone, but not combined with any additional parameters such as command groups, command object types, and arguments.
“^add\s+vserver\s+.*$”	All create virtual server actions, which consist of the add virtual server command followed by a space and more parameters such as command groups, command object types, and arguments.
“^add\s+(lb\s+vserver)\s+.*”	All create lb virtual server actions, which consist of the add lb virtual server command followed by a space and more parameters such as command groups, command object types, and arguments.

The following table shows the command specifications for each of the built-in command policies.

For information about built-in command policies, see table Built-in command policy table.

To create a command policy by using the command line interface

At the command prompt, type the following commands to create a command policy and verify the configuration:

- `add system cmdPolicy <policyname> <action> <cmdspec>`
- `show system cmdPolicy <policyName>`

Example

Example:

```
add system cmdPolicy USER-POLICY ALLOW (\ server\ )|(\ service(Group)*\ )
|(\ vserver\ )|(\ policy\ )|(\ policylabel\ )|(\ limitIdentifier\ )|(^show\
 (?!(system|ns\ (ns.conf|runningConfig))))|(save)|(stat\ .*serv)
```

To configure a command policy by using the GUI

Navigate to **System** > **User Administration** > **Command Policies**, and create the command policy.

Binding command policies to users and groups

Once you have defined your command policies, you must bind them to the appropriate user accounts and groups. When you bind a policy, you must assign it a priority so that the appliance can determine which command policy to follow when two or more applicable command policies are in conflict.

Command policies are evaluated in the following order:

- Command policies bound directly to users and the corresponding groups are evaluated according to priority number. A command policy with a lower priority number is evaluated before one with a higher priority number. Therefore, any privileges the lower-numbered command policy explicitly grants or denies are not overridden by a higher-numbered command policy.
- When two command policies, one bound to a user account and other to a group, have the same priority number, the command policy bound directly to the user account is evaluated first.

To bind command policies to a user by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user and verify the configuration:

- `bind system user <userName> -policyName <policyName> <priority>`
- `show system user <userName>`

Example

```
> bind system user user1 -policyName read_all 1
```

To bind command policies to a user by using the GUI

Navigate to **System** > **User Administration** > **Users**, select the user and bind command policies.

Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

To bind command policies to a group by using the command line interface

At the command prompt, type the following commands to bind a command policy to a user group and verify the configuration:

- `bind system group <groupName> -policyName <policyName> <priority>`

- `show system group <groupName>`

Example

```
> bind system group Managers -policyName read_all 1
```

To bind command policies to a group by using the GUI

Navigate to **System** > **User Administration** > **Groups**, select the group and bind command policies.

Optionally, you can modify the default priority to ensure that the policy is evaluated in the proper order.

Resetting the default administrator password

September 21, 2020

The administrator account provides complete access to all ADC features. Therefore, to preserve security, the administrator account must be used only if necessary and if you require full access to know the password for the administrator account. It is recommended that you frequently change your password. If you lose the password, you must reset it to the default and then change it to a new one.

To reset your password, you must log on to your appliance and change the password. However, if you do not remember the password, you can reboot appliance in single user mode, mount the file system in read/write mode and then remove the **Citrix ADC nsroot** entry from the ns.conf file. As a final step, reboot and log onto your appliance with the default password and set a new password.

To reset the administrator password

1. Connect a computer to the console port of the NetScaler and log on.

   > **Note**
   >
   > You cannot log on by using SSH to perform this procedure; you must connect directly to the appliance.

2. Reboot the NetScaler.
3. Press CTRL+C when the following message appears:

   Press [Ctrl-C] for command prompt, or any other key to boot immediately.

   Booting [kernel] in # seconds.

4. Run the following command to start the NetScaler in a single user mode:

   ```
   boot -s
   ```

> **Note** If boot `-s` does not work, then try reboot `-- -s` and appliance reboot in single user mode.

After the appliance boots, it displays the following message:

Enter full path name of shell or `RETURN for /bin/sh:`

5. Press ENTER key to display the # prompt, and type the following commands to mount the file systems:
    a) Run the following command to check the disk consistency:

    ```
    fsck_ufs /dev/ad0s1a
    ```

    > **Note:**
    >
    > Your flash drive has a specific device name depending on your Citrix ADC; hence, you have to replace ad0s1a in the preceding command with the appropriate device name.

    b) You must access the dev directory and enter 'ls' to check the drive details.

    c) Run the following command to display the mounted partitions:

    ```
    df
    ```

    If the flash partition is not listed, you must mount it manually.

    d) Run the following command to mount the flash drive:

    ```
    mount/dev/ad0s1a /flash
    ```

6. Run the following command to change to the nsconfig directory:

    ```
    cd/flash/nsconfig
    ```

7. Run the following commands to rewrite the ns.conf file and remove the set of system commands defaulting to the administrator user:
    a) Run the following command to create a configuration file that does not have commands defaulting to the administrator:

    ```
    grep -v “set system user nsroot” ns.conf \> new.conf
    ```

    b) Run the following command to make a backup of the existing configuration file:

    ```
    mv ns.conf old.ns.conf
    ```

    c) Run the following command to rename the new.conf file to ns.conf:

    ```
    mv new.conf ns.conf
    ```

8. Run the following command to reboot the NetScaler:

    ```
    reboot
    ```

9. Log on using the default nsroot user credentials.
10. Run the following command to reset the password:

    ```
    set system user nsroot <New_Password>
    ```

    > **Note:**
    >
    > To use the "?" character in a password string, precede this character with the "\" character.

    For example, yourexamplepasswd\? is set for the administrator account after you perform the following operation:

    ```
    > set system user nsroot yourexamplepasswd\?
    ```

Example of a user scenario

September 21, 2020

The following example shows how to create a complete set of user accounts, groups, and command policies and bind each policy to the appropriate groups and users. The company, Example Manufacturing, Inc., has three users who can access the NetScaler appliance:

- **John Doe.** The IT manager. John needs to be able to see all parts of the NetScaler configuration but does not need to modify anything.

  Maria Ramiez. The lead IT administrator. Maria needs to be able to see and modify all parts of the NetScaler configuration except for NetScaler commands (which local policy dictates must be performed while logged on as nsroot).
- **Michael Baldrock.** The IT administrator in charge of load balancing. Michael needs to be able to see all parts of the NetScaler configuration, but needs to modify only the load balancing functions.

The following table shows the breakdown of network information, user account names, group names, and command policies for the sample company.

Field	Value	Note
NetScaler host name	ns01.example.net	N/A
User accounts	johnd, mariar, and michaelb	John Doe, IT manager, Maria Ramirez, IT administrator and Michael Baldrock, IT administrator.

Field	Value	Note
Groups	Managers and SysOps	All managers and all IT administrators.
Command Policies	read_all, modify_lb, and modify_all	Allow complete read-only access, Allow modify access to load balancing, and Allow complete modify access.

The following description walks you through the process of creating a complete set of user accounts, groups, and command policies on the NetScaler appliance named ns01.example.net.

The description includes procedures for binding the appropriate user accounts and groups to one another, and binding appropriate command policies to the user accounts and groups.

This example illustrates how you can use prioritization to grant precise access and privileges to each user in the IT department.

The example assumes that initial installation and configuration have already been performed on the NetScaler.

Configuration steps

1. Use the procedure described in "Configuring User Accounts" to create user accounts **johnd**, **mariar**, and **michaelb**.
2. Use the procedure described in "Configuring User Groups" to create user groups **Managers** and **SysOps**, and then bind the users **mariar** and **michaelb** to the **SysOps** group and the user **johnd** to the **Managers** group.
3. Use the procedure described in "[Creating Custom Command Policies] to create the following command policies:
    - **read_all** with action **Allow** and command spec `"(^show\s+(?!system)(?!ns ns.conf)(?!ns runningConfig).*)|(^stat.*)"`
    - **modify_lb** with action as **Allow** and the command spec `"^set\s+lb\s+.*$"`
    - modify_all with action as Allow and the command spec `"^\S+\s+(?!system).*"`
4. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **read_all** command policy to the **SysOps** group, with priority value **1**.
5. Use the procedure described in "Binding Command Policies to Users and Groups" to bind the **modify_lb** command policy to user **michaelb**, with priority value **5**.

The configuration you just created results in the following:

- John Doe, the IT manager, has read-only access to the entire NetScaler configuration, but he cannot make modifications.
- Maria Ramirez, the IT lead, has near-complete access to all areas of the NetScaler configuration, having to log on only to perform NetScaler-level commands.
- Michael Baldrock, the IT administrator responsible for load balancing, has read-only access to the NetScaler configuration, and can modify the configuration options for load balancing.

The set of command policies that applies to a specific user is a combination of command policies applied directly to the user's account and command policies applied to the group(s) of which the user is a member.

Each time a user enters a command, the operating system searches the command policies for that user until it finds a policy with an ALLOW or DENY action that matches the command. When it finds a match, the operating system stops its command policy search and allows or denies access to the command.

If the operating system finds no matching command policy, it denies the user access to the command, in accordance with the NetScaler appliance's default deny policy.

> **Note**
>
> When placing a user into multiple groups, take care not to cause unintended user command restrictions or privileges. To avoid these conflicts, when organizing your users in groups, bear in mind the NetScaler command policy search procedure and policy ordering rules.

TCP Configurations

April 20, 2021

TCP configurations for a NetScaler appliance can be specified in an entity called a TCP profile, which is a collection of TCP settings. The TCP profile can then be associated with services or virtual servers that want to use these TCP configurations.

A default TCP profile can be configured to set the TCP configurations that will be applied by default, globally to all services and virtual servers.

> **Note**
>
> When a TCP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence. The NetScaler appliance also provides other approaches for configuring TCP.

Supported TCP configuration

The NetScaler appliance supports the following TCP capabilities:

Defending TCP against spoofing attacks

The NetScaler implementation of window attenuation is RFC 4953 compliant.

Explicit Congestion Notification (ECN)

The appliance sends notification of the network congestion status to the sender of the data and takes corrective measures for data congestion or data corruption. The NetScaler implementation of ECN is RFC 3168 compliant.

Round trip time measurement (RTTM) using the timestamp option

For the TimeStamp option to work, at least one side of the connection (client or server) must support it. The NetScaler implementation of TimeStamp option is RFC 1323 compliant.

Detection of spurious retransmissions

This can be done using TCP duplicate selective acknowledgement (D-SACK) and forward RTO-Recovery (F-RTO). In case of spurious retransmissions, the congestion control configurations are reverted to their original state. The NetScaler implementation of D-SACK is RFC 2883 compliant, and F-RTO is RFC 5682 compliant.

Congestion control

This functionality use New-Reno, BIC, CUBIC, Nile and TCP Westwood algorithms.

Window scaling

This increases the TCP receive window size beyond its maximum value of 65,535 bytes.

Points to consider before you configure window scaling

- You do not set a high value for the scale factor, because this could have adverse effects on the appliance and the network.
- You do not configure window scaling unless you clearly know why you want to change the window size.
- Both hosts in the TCP connection send a window scale option during connection establishment. If only one side of a connection sets this option, window scaling is not used for the connection.

- Each connection for same session is an independent window scaling session. For example, when a client's request and the server's response flow through the appliance, it is possible to have window scaling between the client and the appliance without window scaling between the appliance and the server.

TCP maximum congestion window

The window size is a user configurable one. The default value is 8190 bytes.

Selective acknowledgment (SACK)

This uses the data receiver (either a NetScaler appliance or a client) notifies the sender about all the segments that have been received successfully.

Forward acknowledgment (FACK)

This functionality avoids TCP congestion by explicitly measuring the total number of data bytes outstanding in the network, and helping the sender (either a NetScaler or a client) control the amount of data injected into the network during retransmission timeouts.

TCP connection multiplexing

This functionality enables reuse of existing TCP connections. The NetScaler appliance stores established TCP connections to the reuse pool. Whenever a client request is received, appliance checks for an available connection in the reuse pool and serves the new client if the connection is available. If it is unavailable, the appliance creates a new connection for the client request and stores the connection to the reuse pool. NetScaler supports connection multiplexing for HTTP, SSL, and DataStream connection types.

Dynamic receive buffering

This allows the receive buffer to be adjusted dynamically based on memory and network conditions.

MPTCP Connection

MPTCP connections between client and NetScaler. MPTCP connections are not supported between NetScaler and the backend server. The NetScaler implementation of MPTCP is RFC 6824 compliant.

You can view MPTCP statistics such as active MPTCP connections and active subflow connections by using the command line interface.

At the command prompt, type one of the following commands to display a summary or detailed summary of MPTCP statistics, or to clear the statistics display:

1. Stat MPTCP
2. Stat mptcp –detail
3. Clearstats basic

> **Note**
>
> To establish an MPTCP connection, both the client and the NetScaler appliance must support the same MPTCP version. If you use the NetScaler appliance as an MPTCP gateway for your servers, the servers do not have to support MPTCP. When the client starts a new MPTCP connection, the appliance identifies the client's MPTPC version from the MP_CAPABALE option in the SYN packet. If the client's version is higher than the one supported on the appliance, the appliance indicates its highest version in the MP_CAPABALE option of the SYN-ACK packet. The client then falls back to a lower version and sends the version number in the MP_CAPABALE option of the ACK packet. If that version is supportable, the appliance continues the MPTCP connection. Otherwise, the appliance falls back to a regular TCP. The NetScaler appliance does not initiate subflows (MP_JOIN's). The appliance expects the client to initiate subflows.

Extracting the TCP/IP path overlay option and inserting client-IP HTTP header

Extracting TCP/IP path overlay and inserting client-IP HTTP header. Data transport through overlay networks often uses connection termination or Network Address Translation (NAT), in which the IP address of the source client is lost. To avoid this, the NetScaler appliance extracts the TCP/IP path overlay option and inserts the source client's IP address into the HTTP header. With the IP address in the header, the web server can identify the source client that made the connection. The extracted data is valid for lifetime of the TCP connection and therefore, this prevents the next hop host from having to interpret the option again. This option is applicable only for web services that have the client-IP insertion option enabled.

Additionally, NetScaler provides configuration support for the following:

- TCP segmentation offload.
- Synchronizing cookie for TCP handshake with clients.
- Learning MSS to enable MSS learning for all the virtual servers configured on the appliance.

Setting Global TCP Parameters

The NetScaler appliance allows you to specify values for TCP parameters that are applicable to all NetScaler services and virtual servers. This can be done using:

- Default TCP profile
- Global TCP command
- TCP buffering feature

> **Note**:
>
> The recvBuffSize parameter of the set ns tcpParam command is deprecated from release 9.2 onwards. In later releases, set the buffer size by using the bufferSize parameter of the set ns tcpProfile command. If you upgrade to a release where the recvBuffSize parameter is deprecated, the bufferSize parameter is set to its default value.

Default TCP profile

A TCP profile, named as nstcp_default_profile, is used to specify TCP configurations that will used if no TCP configurations are provided at the service or virtual server level.

> **Notes**:
>
> - Not all TCP parameters can be configured through the default TCP profile. Some settings have to be performed by using the global TCP command (see section below).
> - The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default TCP profile

- Using the command line interface, at the command prompt enter:

  ```
  set ns tcpProfile nstcp_default_profile …
  ```

- On the GUI, navigate to **System** > **Profiles**, click **TCP Profiles** and update nstcp_default_profile.

Global TCP command

Another approach you can use to configure global TCP parameters is the global TCP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using a TCP profile. Any update made to these duplicate parameters is reflected in the corresponding parameter in the default TCP profile.

For example, if the SACK parameter is updated using this approach, the value is reflected in the SACK parameter of the default TCP profile (nstcp_default_profile).

> **Note**:
>
> Citrix recommends that you use this approach only for TCP parameters that are not available in the default TCP profile.

To configure the global TCP command

- Using the command line interface, at the command prompt enter:

```
set ns tcpParam …
```

- On the GUI, navigate to **System** > **Settings**, click **Change TCP parameters** and update the required TCP parameters.

TCP buffering feature

NetScaler provides a feature called TCP buffering that you can use to specify the TCP buffer size. The feature can be enabled globally or at service level.

> **Note**
> The buffer size can also be configured in the default TCP profile. If the buffer size has different values in the TCP buffering feature and the default TCP profile, the greater value is applied.

To configure the TCP buffering feature globally

- At the command prompt enter:

  enable ns mode TCPB

```
set ns tcpbufParam -size <positiveInteger> -memLimit <positiveInteger>
```

- On the GUI, navigate to **System** > **Settings**, click **Configure Modes** and select **TCP Buffering**.

  And, navigate to **System** > **Settings**, click **Change TCP parameters** and specify the values for **Buffer size** and **Memory usage limit**.

Setting Service or Virtual Server Specific TCP Parameters

Using TCP profiles, you can specify TCP parameters for services and virtual servers. You must define a TCP profile (or use a built-in TCP profile) and associate the profile with the appropriate service and virtual server.

> **Note**:
>
> - You can also modify the TCP parameters of default profiles as per your requirements.
> - You can specify the TCP buffer size at service level using the parameters specified by the TCP buffering feature.

To specify service or virtual server level TCP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the TCP profile.

```
set ns tcpProfile <profile-name>...
```

1. Bind the TCP profile to the service or virtual server.

`set service <name> ....`

Example:

`> set service service1 -tcpProfileName profile1`

```
To bind the TCP profile to the virtual server:
```

`set lb vserver <name> ....`

Example:

`> set lb vserver lbvserver1 -tcpProfileName profile1`

```
To specify service or virtual server level TCP configurations by using
    the GUI
```

At the GUI, perform the following:

1. Configure the TCP profile.

   Navigate to **System** > **Profiles** > **TCP Profiles**, and create the TCP profile.

2. Bind the TCP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing** > **Services/Virtual Servers**, and create the TCP profile, which should be bound to the service or virtual server.

Built-in TCP Profiles

For convenience of configuration, the NetScaler provides some built-in TCP profiles. Review the built-in profiles listed below and select a profile and use it as it is or modify it to meet your requirements. You can bind these profiles to your required services or virtual servers.

Built-in profile	Description
nstcp_default_profile	Represents the default global TCP settings on the appliance.
nstcp_default_tcp_lan	Useful for back-end server connections, where these servers reside on the same LAN as the appliance.
nstcp_default_WAN	useful for WAN deployments.
nstcp_default_tcp_lan_thin_stream	Similar to the nstcp_default_tcp_lan profile; however, the settings are tuned to small size packet flows.

Built-in profile	Description
nstcp_default_tcp_interactive_stream	Similar to the nstcp_default_tcp_lan profile; however, it has a reduced delayed ACK timer and ACK on PUSH packet settings.
nstcp_default_tcp_lfp	Useful for long fat pipe networks (WAN) on the client side. Long fat pipe networks have long delay, high bandwidth lines with minimal packet drops.
nstcp_default_tcp_lfp_thin_stream	Similar to the nstcp_default_tcp_lfp profile; however, the settings are tuned for small size packet flows.
nstcp_default_tcp_lnp	Useful for long narrow pipe networks (WAN) on the client side. Long narrow pipe networks have considerable packet loss once in a while.
nstcp_default_tcp_lnp_thin_stream	Similar to the nstcp_default_tcp_lnp profile; however, the settings are tuned for small size packet flows.
nstcp_internal_apps	Useful for internal applications on the appliance (for example, GSLB sitesyncing). This contains tuned window scaling and SACK options for the desired applications. This profile should not be bound to applications other than internal applications.
nstcp_default_Mobile_profile	Useful for mobile devices.
nstcp_default_XA_XD_profile	Useful for a XenApp or XenDesktop deployment.

Sample TCP configurations

Sample command line interface examples for configuring the following:

Defending TCP against spoofing attacks

Enable the NetScaler to defend TCP against spoof attacks. By default the "rstWindowAttenuation" parameter is disabled. This parameter is enabled to protect the appliance against spoofing. If you enable, it will reply with corrective acknowledgement (ACK) for an invalid sequence number. Possible values are Enabled, Disabled.

Where RST window attenuate parameter protects the appliance against spoofing. When enabled, will reply with corrective ACK when a sequence number is invalid.

~~~
```
> set ns tcpProfile profile1 -rstWindowAttenuate ENABLED -spoofSynDrop ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Explicit congestion notification (ECN)

`Enable ECN on the required TCP profile`

~~~
```
> set ns tcpProfile profile1 -ECN ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Selective ACKnowledgment (SACK)

Enable SACK on the required TCP profile.

~~~
```
> set ns tcpProfile profile1 -SACK ENABLED
Done
`> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Forward ACKnowledgment (FACK)

Enable FACK on the required TCP profile.

~~~
```
> set ns tcpProfile profile1 -FACK ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Window Scaling (WS)

Enable window scaling and set the window scaling factor on the required TCP profile.

~~~
```
> set ns tcpProfile profile1 – WS ENABLED – WSVal 9
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Maximum Segment Size (MSS)

Update the MSS related configurations.

~~~
```
> set ns tcpProfile profile1 – mss 1460 - maxPktPerMss 512
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

NetScaler to learn the MSS of a virtual server

Enable the NetScaler to learn the VSS and update other related configurations.

~~~
```
> set ns tcpParam -learnVsvrMSS ENABLED – mssLearnInterval 180 -mssLearnDelay 3600
Done
<!--NeedCopy--> ```
~~~

TCP keep-alive

Enable TCP keep-alive and update other related configurations.

~~~
```
> set ns tcpProfile profile1 – KA ENABLED – KaprobeUpdateLastactivity ENABLED -KAconnIdleTime 900 -KAmaxProbes 3 -KaprobeInterval 75
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Buffer size - using TCP profile

Specify the buffer size.

~~~
```
> set ns tcpProfile profile1  - bufferSize 8190
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Buffer size - using TCP buffering feature

Enable the TCP buffering feature (globally or for a service) and then specify the buffer size and the memory limit.

~~~
```
> enable ns feature TCPB
Done
> set ns tcpbufParam -size 64 -memLimit 64
Done
<!--NeedCopy--> ```
~~~

MPTCP

Enable MPTCP and then set the optional MPTCP configurations.

~~~
```
> set ns tcpProfile profile1 -mptcp ENABLED
Done
> set ns tcpProfile profile1 -mptcpDropDataOnPreEstSF ENABLED -mptcpFastOpen ENABLED -mptcpSessionTimeout 7200
Done
> set ns tcpparam -mptcpConCloseOnPassiveSF ENABLED -mptcpChecksum ENABLED -mptcpSFtimeout 0 -mptcpSFReplaceTimeout 10
-mptcpMaxSF 4 -mptcpMaxPendingSF 4 -mptcpPendingJoinThreshold 0 -mptcpRTOsToSwitchSF 2 -mptcpUseBackupOnDSS ENABLED
Done
<!--NeedCopy--> ```
~~~

Congestion control

Set the required TCP congestion control algorithm.

~~~
```
> set ns tcpProfile profile1 -flavor Westwood
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Dynamic receive buffering

Enable dynamic receive buffering on the required TCP profile.

~~~
```
> set ns tcpProfile profile1 -dynamicReceiveBuffering ENABLED
Done
> set lb vserver lbvserver1 -tcpProfileName profile1
Done
<!--NeedCopy--> ```
~~~

Support for TCP Fast Open (TFO) in Multipath TCP (MPTCP)

A NetScaler appliance now supports TCP Fast Open (TFO) mechanism for establishing Multipath TCP (MPTCP) connections and speed up data transfers. The mechanism allows subflow data to be carried during the initial MPTCP connection handshake in SYN and SYN-ACK packets and also enables data to be consumed by the receiving node during the MPTCP connection establishment.

For more information, see TCP Fast Open topic.

Support for Variable TFO Cookie Size for MPTCP

A NetScaler appliance now enables you to configure a variable length TCP Fast Open (TFO) cookie of a minimum size of 4 bytes and a maximum size of 16 bytes in a TCP profile. By doing this, the appliance can respond with the configured TFO cookie size in the SYN-ACK packet to the client.

To configure TCP Fast Open (TFO) cookie in a TCP profile by using the command line interface

At the command prompt, type:

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize <positive_integer
>
```

Example

```
set tcpProfile nstcp_default_profile -tcpFastOpenCookieSize 8
```

To configure TCP Fast Open (TFO) cookie in a TCP profile by using GUI

1. Navigate to **Configuration** > **System** > Profiles.
2. In the details pane, go to **TCP Profiles** tab and select a TCP profile.
3. In the **Configure TCP Profile** page, set the **TCP Fast Open** cookie size.
4. Click **OK** and **Done**.

SYN-Cookie timeout interval

In a NetScaler appliance, by default, the SYN cookie parameter on the TCP profile is enabled to resist SYN attacks. If you prefer to detect an attack for a virtual server and check the SYN-ACK retransmission rate, the SYN cookie value is toggled from Enabled to Disabled state. However, the toggling effect of the cookie from Enabled to Disabled state and vice versa causes a configuration inconsistency between nodes in a cluster deployment. To overcome this, you can use the second SYN cookie parameter maintained on each virtual server of the appliance. When a SYN attack is detected, the SYN cookie on the TCP profile is disabled and the second SYN cookie value on the server is toggled from Disabled to Enabled state. The cookie remains in Enabled state for a autosyncookietimeout interval that you can specify. During the timeout interval, the virtual server fights the SYN attack in SYN Cookie mode. After the interval times out, the SYN Cookie value is toggled back to Disabled state if the virtual server's SYN-ACK retransmission rate is below the threshold. Otherwise, if the rate is still above the threshold, it implies it is a SYN attack and the cookie is again toggled to Enabled state for the specified timeout period.

To configure the maximum SYN ACK retransmission threshold by using the command line interface

At the command prompt, type:

~~~~
```
set ns tcpparam [-maxSynAckRetx <positive_integer>]

Set ns tcpparam [-maxSynAckRetx 150]
<!--NeedCopy--> ```
~~~~

To configure auto SYN cookie timeout interval by using the command line interface

At the command prompt, type:

```
set ns tcpparam [-autosyncookietimeout <positive_integer>]
```

```
Set ns tcpparam [-autosyncookietimeout 90]
```

HTTP configurations

November 5, 2020

HTTP configurations for a NetScaler appliance can be specified in an entity called an HTTP profile, which is a collection of HTTP settings. The HTTP profile can then be associated with services or virtual servers that want to use these HTTP configurations.

A default HTTP profile can be configured to set the HTTP configurations that will be applied by default, globally to all services and virtual servers.

> **Note:** When a HTTP parameter has different values for service, virtual server, and globally, the value of the most-specific entity (the service) is given the highest precedence.

The NetScaler appliance also provides other approaches for configuring HTTP. Read on for more information.

The NetScaler supports
WebSocket protocol which allows browsers and other clients to create a bi-directional, full duplex TCP connection to the servers. The NetScaler implementation of WebSocket is RFC
6455 compliant.

> **Note:**
>
> A NetScaler appliance now supports the User Source IP (USIP) address configuration for both HTTP/1.1 and HTTP/2 protocols.

Setting global HTTP parameters

The NetScaler appliance allows you to specify values for HTTP parameters that are applicable to all NetScaler services and virtual servers. This can be done using:

- Default HTTP profile
- Global HTTP command

Default HTTP profile

A HTTP profile, named as `nshttp_default_profile`, is used to specify HTTP configurations that will be used if no HTTP configurations are provided at the service or virtual server level.

> **Note:**
>
> - Not all HTTP parameters can be configured through the default HTTP profile. Some settings have to be performed by using the global HTTP command (see section below).
> - The default profile does not have to be explicitly bound to a service or virtual server.

To configure the default HTTP profile

- Using the command line interface, at the command prompt enter:

  ```
  set ns httpProfile nshttp_default_profile …
  ```

- On the GUI, navigate to **System** > **Profiles**, click **HTTP Profiles** and update nshttp_default_profile.

Global HTTP command

Another approach you can use to configure global HTTP parameters is the global HTTP command. In addition to some unique parameters, this command duplicates some parameters that can be set by using a HTTP profile. Any update made to these duplicate parameters is reflected in the corresponding parameter in the default HTTP profile.

For example, if the maxReusePool parameter is updated using this approach, the value is reflected in the maxReusePool parameter of the default HTTP profile (nshttp_default_profile).

> **Note:**
>
> Citrix recommends that you use this approach only for HTTP parameters that are not available in the default HTTP profile.

To configure the global HTTP command

- Using the command line interface, at the command prompt enter:

  ```
  set ns httpParam …
  ```

- On the GUI, navigate to System > Settings, click Change HTTP parameters and update the required HTTP parameters.

To configure the ignore Coding scheme for connect request

To enable HTTP/2 and set HTTP/2 parameters to ignore the Coding scheme in the connect request, at the command prompt, type:

```
set ns httpParam [-ignoreConnectCodingScheme ( ENABLED | DISABLED )]
```

Example:

```
set ns httpParam -ignoreConnectCodingScheme ENABLED
```

Setting Service or Virtual Server Specific HTTP Parameters

Using HTTP profiles, you can specify HTTP parameters for services and virtual servers. You must define a HTTP profile (or use a built-in HTTP profile) and associate the profile with the appropriate service and virtual server.

> **Note:**
>
> You can also modify the HTTP parameters of default profiles as per your requirements.

To specify service or virtual server level HTTP configurations by using the command line interface

At the command prompt, perform the following:

1. Configure the HTTP profile.

   `set ns httpProfile <profile-name>...`

2. Bind the HTTP profile to the service or virtual server.

   To bind the HTTP profile to the service:

`set service <name> .....`

```
**Example:**

`> set service service1 -httpProfileName profile1`

To bind the HTTP profile to the virtual server:

`set lb vserver \<name\> .....`

**Example:**

`> set lb vserver lbvserver1 -httpProfileName profile1`
```

To specify service or virtual server level HTTP configurations by using the GUI

At the GUI, perform the following:

1. Configure the HTTP profile.

   Navigate to **System** > **Profiles** > **HTTP Profiles**, and create the HTTP profile.

2. Bind the HTTP profile to the service or virtual server.

   Navigate to **Traffic Management** > **Load Balancing** > **Services/Virtual Servers**, and create the HTTP profile, which should be bound to the service or virtual server.

Built-in HTTP Profiles

For convenience of configuration, the NetScaler provides some built-in HTTP profiles. Review the profiles listed below and use it as it is or modify it to meet your requirements. You can bind these profiles to the required services or virtual servers.

Built-in profile	Description
nshttp_default_profile	Represents the default global HTTP settings on the appliance.
nshttp_default_strict_validation	Settings for deployments that require strict validation of HTTP requests and responses.

Sample HTTP Configurations

Sample command line interface examples to configure the following:

- HTTP band statistics
- WebSocket connections

HTTP band statistics

Specify the band size for HTTP requests and responses.

```
> set protocol httpBand reqBandSize 300 respBandSize 2048
> show protocol httpband -type REQUEST
```

WebSocket connections

Enable webSocket on the required HTTP profile.

```
> set ns httpProfile http_profile1 -webSocket ENABLED
> set lb vserver lbvserver1 -httpProfileName profile1
```

Configuring HTTP/2 on the NetScaler appliance

September 21, 2020

Note: The HTTP/2 functionality is supported on the NetScaler MPX, VPX, and SDX models. For NetScaler VPX, the HTTP/2 functionality is supported from 11.0 release onwards.

The problem with web application performance is directly related to the trend toward increasing the page size and the number of objects on the web pages. HTTP/1.1 was developed to support smaller web pages, slower Internet connections, and more limited server hardware than are common today. It is not well suited for newer technologies such as JavaScript and cascading style sheets (CSS), nor for new media types such as Flash videos and graphics-rich images, because it can request only one resource per connection to the server. This limitation significantly increases the number of round trips, causing longer page-rendering time and reduced network performance.

The HTTP/2 protocol addresses these limitations by allowing communication to occur with less data transmitted over the network, and providing the ability to send multiple requests and responses across a single connection. At its core, HTTP/2 addresses the key limitations of HTTP/1.1 by using the underlying network connections more efficiently. It changes the way requests and responses travel over the network.

HTTP/2 is a binary protocol. It is more efficient to parse, more compact on the wire, and most importantly, it is less error-prone, compared to textual protocols like HTTP/1.1. The HTTP/2 protocol uses a binary framing layer that defines the frame type and how HTTP messages are encapsulated and transferred between the client and server.The HTTP/2 functionality supports the use of the CONNECT method to establish a tunnel connection through a single HTTP/2 stream to a remote host.

The HTTP/2 protocol includes a lot of performance-enhancing changes that significantly improve performance, particularly for clients connecting over a mobile network.

The following table lists the major improvements in HTTP/2 over HTTP/1.1:

HTTP/2 Features	**Description**
Header Compression	HTTP headers have a lot of repetitive information and therefore consume unnecessary bandwidth during data transmission. HTTP/2 reduces bandwidth requirements by compressing the header and minimizing the requirement to transport HTTP headers with every request and response.
Connection Multiplexing	Latency can have a huge impact on page load times and the end user experience. Connection multiplexing overcomes this problem by sending multiple requests and responses across a single connection.
Server Push	Server push enables the server to proactively push content to the client browser, avoiding round trip delay. This feature caches the responses it thinks the client will need, reduces the number round trips, and improves the page rendering time. Important: The NetScaler appliance does not support the server push functionality.

HTTP/2 Features	Description
No Head-of-line Blocking	Under HTTP/1.1, browsers can download one resource at a time per connection. When a browser has to download a large resource, it blocks all other resources from downloading until the first download is complete. HTTP/2 overcomes this problem with a multiplexing approach. It allows the client browser to download other web components in parallel over the same connection and display them as they become available.
Request Prioritization	Not all resources have equal priority when the browser renders a web page. To accelerate the load time, all modern browsers prioritize requests by type of asset, their location on the page, and even by learned priority from previous visits. With HTTP/1.1, the browser has limited ability to leverage the priority data, because this protocol does not support multiplexing, and there is no way to communicate request prioritization by the server. The result is unnecessary network latency. HTTP/2 overcomes this problem by allowing the browser to dispatch all requests. The browser can communicate its stream prioritization preference via stream dependencies and weights, enabling the servers to optimize response delivery. Important: The NetScaler appliance does not support the request prioritization functionality.

How HTTP/2 works on a NetScaler appliance

A NetScaler appliance supports HTTP/2 on the client side as well on the server side. On the client side, the NetScaler appliance acts as a server that hosts an HTTP/HTTPS virtual server for HTTP/2. On the backend side, the NetScaler acts as a client to the servers that are bound to the virtual server.

Therefore, the NetScaler appliance maintains separate connections on the client side as well on the server side. The NetScaler appliance has separate HTTP/2 configurations for the client side and the server side.

HTTP/2 for HTTPS (SSL) load balancing configuration

For an HTTPS load balancing configuration, the NetScaler appliance uses the TLS ALPN extension (RFC 7301) to determine whether the client/server supports HTTP/2. If it does, the appliance chooses HTTP/2 as the application-layer protocol to transmit data (as described in RFC 7540 - Section 3.3) on the client/server side.
The appliance uses the following order of preference when choosing the application-layer protocol through the TLS ALPN extension:

- HTTP/2 (if enabled in the HTTP profile)
- SPDY (if enabled in the HTTP profile)
- HTTP/1.1

HTTP/2 for HTTP load balancing configuration

For an HTTP load balancing configuration, the NetScaler appliance uses one of the following methods to start communicating with the client/server using HTTP/2.

> **Note**
>
> In the following method descriptions, client and server are generals terms for an HTTP/2 connection. For example, for a load balancing setup of a NetScaler appliance using HTTP/2, the NetScaler appliance acts as a server on the client side and acts as a client to the server side.
>
> If the HTTP/2 setting is enabled on the HTTP profile associated with the virtual server, the NetScaler appliance does *NOT* support the direct use of HTTP/2,
> but still it acts as a TCP proxy by forwarding the HTTP/2 requests to servers.

Configuring HTTP/2

The HTTP/2 feature is part of the HTTP configurations on the NetScaler appliance. By default, the HTTP/2 feature is disabled. You have to enable the feature on a
specific HTTP profile and associate the HTTP profile with services or virtual servers on which you want to use the HTTP/2 feature.

To know more about binding a HTTP profile to a virtual server, see Setting Service or Virtual Server Specific HTTP Parameters section on HTTP Configuration page.

> **Note**
>
> The HTTP/2 functionality will not work, if User Source IP (USIP) mode is enabled and the Proxy mode is disabled on the NetScaler appliance.

To configure HTTP/2 by using the Citrix ADC GUI

1. Create a HTTP/2 profile on the NetScaler appliance. Navigate to System > Profiles, click **HTTP Profiles** and then click **Add** to create a HTTP profile
   with HTTP/2 checkbox selected.
2. [Optional] Configure HTTP/2 header table size.
   Navigate to **System** > **Profiles**, select a HTTP Profile and click **Edit** to specify the maximum size of the header compression table in the **HTTP/2 Header Table Size** text box.
3. [Optional] Configure HTTP/2 initial window size
   Navigate to **System** > **Profiles**, select a HTTP Profile and click **Edit** to specify initial window size for an HTTP/2 stream in the the **HTTP/2 Initial Window Size** text box.
4. [Optional] Configure HTTP/2 maximum concurrent streams.

   Navigate to **System** > **Profiles**, select a HTTP Profile and click Edit to specify the maximum number of concurrent streams per connection in the HTTP/2
   Maximum Concurrent Streams text box .
5. [Optional] Configure HTTP/2 maximum frame size.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click Edit to specify the maximum size of the frame in the **HTTP/2 Maximum Frame Size** text box.
6. [Optional] Configure HTTP/2 maximum header list size.
   Navigate to **System** > **Profiles**, select a HTTP Profiles and click **Edit** to specify the header size the **Maximum Header List Size** text box.

SNMP

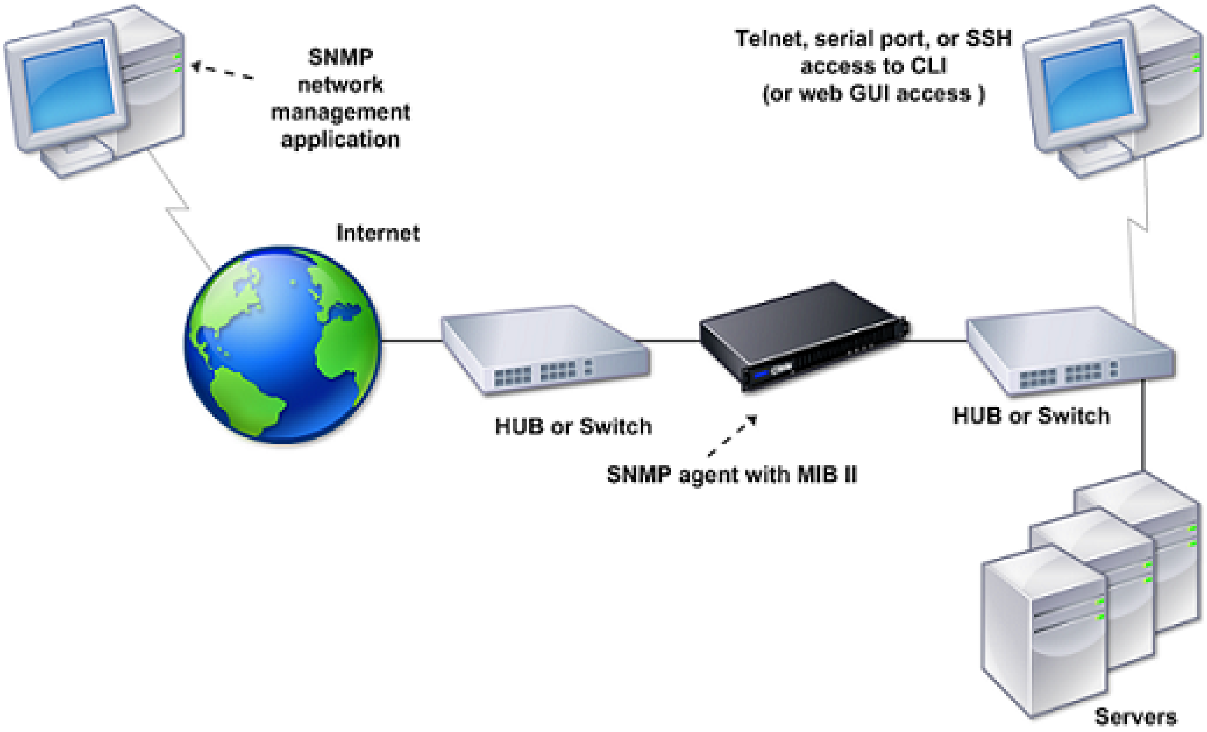
July 30, 2021

You can use Simple Network Management Protocol (SNMP) to configure the SNMP agent on the NetScaler appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the NetScaler. The traps are then sent to a remote device called a *trap listener*, which signals the abnormal condition on the NetScaler appliance. Or, you can query the SNMP agent for System-specific information from a remote device called an *SNMP manager*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The SNMP agent on the NetScaler can generate traps compliant with SNMPv1, SNMPv2, and SNMPv3. For querying, the SNMP agent supports SNMP version 1 (SNMPv1), SNMP version 2 (SNMPv2), and SNMP version 3 (SNMPv3).

For information about SNMP parameters, traps, and its descriptions, see NetScaler SNMP OID Reference.

The following figure illustrates a network with a NetScaler that has SNMP enabled and configured. In the figure, each SNMP network management application uses SNMP to communicate with the SNMP agent on the NetScaler. The SNMP agent searches its management information base (MIB) to collect the data requested by the SNMP Manager and provides the information to the application.


SNMP network management application
Telnet, serial port, or SSH access to CLI (or web GUI access)
Internet
HUB or Switch
HUB or Switch
SNMP agent with MIB II
Servers

> **Important**
>
> The SNMP module in a NetScaler appliance supports a maximum length of 128 bytes (as compliant with RFC 3416) for an SNMP OID. A long index variable name for an object can result in an SNMP OID exceeding 128 bytes in length.
>
> To resolve this issue, the NetScaler SNMP module supports a maximum length of 31 characters for an index variable name. If an index variable name exceeds 31 characters in length, the SNMP module using an hash algorithm converts the name to a 31 characters hash value. This hashed value is used in the SNMP OID for that variable.
>
> The original index variable name is stored in another variable, which has the following name format: `<variable type>FullName`. For example, When the name of a load balancing vir-

> tual server has more the 31 characters, `vserverName` SNMP OID contains the hashed value and `vsvrFullName` SNMP OID contains the full (original) name of the virtual server.
>
> Similarly, for SNMP traps, the index variable displays a hashed valued. `<variable type>FullName`, which stores the full name of the original index variable name, is also part of the trap messages.

Importing MIB Files to the SNMP Manager and Trap Listener

To monitor a NetScaler appliance, you must download the MIB object definition files. The NetScaler appliance supports the following enterprise-specific MIBs:

- **A subset of standard MIB-2 groups**. Provides MIB-2 groups SYSTEM, IF, ICMP, UDP, and SNMP.
- **A system enterprise MIB**. Provides system-specific configuration and statistics.

You can obtain the MIB object definition files from the /netscaler/snmp directory or from the Downloads tab of the GUI.

Configuring the NetScaler to generate SNMP traps

September 21, 2020

You can configure the NetScaler appliance to generate asynchronous events, which are called *traps*. The traps are generated whenever there are abnormal conditions on the appliance. The traps are sent to a remote device called a *trap listener*. This helps administrators monitor the appliance and respond promptly to any issues.

The NetScaler appliance provides a set of condition entities called *SNMP alarms*. When the condition in any SNMP alarm is met, the appliance generates SNMP trap messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

To configure the NetScaler appliance to generate traps, you need to enable and configure alarms. Then, you specify trap listeners to which the appliance will send the generated trap messages.

Enabling an SNMP alarm

The NetScaler appliance generates traps only for SNMP alarms that are enabled. Some alarms are enabled by default, but you can disable them.

When you enable an SNMP alarm, the appliance generates corresponding trap messages when some events occur. Some alarms are enabled by default.

To enable an SNMP alarm by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `enable snmp alarm <trapName>`
- `show snmp alarm <trapName>`

To enable an SNMP alarm by using the GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select the alarm.
2. Click **Actions** and select **Enable**.

Configuring alarms

The NetScaler appliance provides a set of condition entities called *SNMP alarms*. When the condition set for an SNMP alarm is met, the appliance generates SNMP traps messages that are sent to the configured trap listeners. For example, when the LOGIN-FAILURE alarm is enabled, a trap message is generated and sent to the trap listener whenever there is a login failure on the appliance.

You can assign an SNMP alarm with a severity level. When you do this, the corresponding trap messages are assigned that severity level.

The following are the severity levels, defined on the appliance, in decreasing order of severity.

- Critical
- Major
- Minor
- Warning
- Informational

For example, if you set a warning severity level for the SNMP alarm named LOGIN-FAILURE, the trap messages generated when there is a login failure will be assigned with the warning severity level.

You can also configure an SNMP alarm to log the corresponding trap messages generated whenever the condition on that alarm is met.

To configure an SNMP alarm by using the command line interface

At the command prompt, type the following commands to configure an SNMP alarm and verify the configuration:

- `set snmp alarm <trapName> [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-time <secs>] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`

- `show snmp alarm <trapName>`

Where,

ThresholdValue: Value for the high threshold. The NetScaler appliance generates an SNMP trap message when the value of the attribute associated with the alarm is greater than or equal to the specified high threshold value.

NormalValue: Value for the normal threshold. A trap message is generated if the value of the respective attribute falls to or below this value after exceeding the high threshold.

To configure SNMP alarms by using the GUI

Navigate to System > SNMP > Alarms, select an alarm and configure the alarm parameters.

Configuring SNMPv1 or SNMPv2 traps

After configuring the alarms, you need to specify the trap listener to which the appliance sends the trap messages. Apart from specifying parameters such as IP or IPv6 address and the destination port of the trap listener, you can specify the type of trap (either generic or specific) and the SNMP version.

You can configure a maximum of 20 trap listeners for receiving either generic or specific traps.

You can also configure the appliance to send SNMP trap messages with a source IP address other than the NetScaler IP (NSIP or NSIP6) address to a particular trap listener. For a trap listener that has an IPv4 address, you can set the source IP to either a mapped IP (MIP) address or a subnet IP (SNIP) address configured on the appliance. For a trap listener that has an IPv6 address, you can set the source IP to subnet IPv6 (SNIP6) address configured on the appliance.

You can also configure the appliance to send trap messages to a trap listener on the basis of a severity level. For example, if you set the severity level as Minor for a trap listener, all trap messages of the severity level equal to or greater than Minor (Minor, Major, and Critical) are sent to the trap listener.

If you have defined a community string for the trap listener, you must also specify a community string for each trap that is to be sent to the listener. A trap listener for which a community string has been defined accepts only trap messages that include a community string matching the community string defined in the trap listener. Other trap messages are dropped.

To add an SNMP trap by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 )-destPort <port> -communityName <string> -srcIP <ip_addr> -severity <severity>`

- `show snmp trap`

Example:

```
> add snmp trap specific 10.102.29.3 -version V2 -destPort 80 -
   communityName com1 -severity Major
<!--NeedCopy-->
```

To configure SNMP traps by using the GUI

Navigate to **System** > **SNMP** > **Traps**, and create the SNMP trap.

Configuring SNMPv3 traps

SNMPv3 provides security capabilities such as authentication and encryption by using the credentials of SNMP users. An SNMP manager can receive SNMPv3 trap messages only if its configuration includes the password assigned to the SNMP user.

The trap destination can now receive SNMPv1, SNMPv2, and SNMPv3 trap messages.

To configure an SNMPv3 trap by using the command line interface

At the command prompt, do the following:

1. Add an SNMPv3 trap.

   `add snmp trap <trapClass> <trapDestination> -version ( V1 | V2 | V3) -destPort <port> -communityName <string> -srcIP <ip_addr> -severity < severity>`

   > **Note:**
   >
   > Once set, the SNMP trap version cannot not be modified.

   Example

   ```
   > add snmp trap specific 10.102.29.3 -version V3 -destPort 80 -
      communityName com1 -severity Major
   <!--NeedCopy-->
   ```

2. Add an SNMP user.

   `add snmp user <name> -group <string> [ -authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`

   Example:

```
     > add snmp user edocs_user -group edocs_group
<!--NeedCopy-->
```

1. Bind the SNMPv3 trap to the SNMP user.

```
bind snmp trap <trapClass> <trapDestination> [-version <version>] (-userName
 <string> [-securityLevel <securityLevel>])
```

Example:

```
     > bind snmp trap specific 10.102.29.3 -version V3 -userName edocs_user -securityLevel authPriv
<!--NeedCopy-->
```

To configure an SNMPv3 trap by using the GUI

1. Add an SNMPv3 trap.

   Navigate to **System** > **SNMP** > **Traps**, and create the SNMP trap by selecting V3 as the SNMP version.

2. Add an SNMP user.

   Navigate to **System** > **SNMP** > **Users** and create the SNMP user.

3. Bind the SNMPv3 trap to the SNMP user.

   - Navigate to **System** > **SNMP** > **Traps**, and select the SNMP version 3 trap.
   - Select the user to which the trap should be bound and define the appropriate Security Level.

SNMP trap logging

A NetScaler appliance can log SNMP trap messages (for SNMP alarms in which logging capability is enabled) when you enable the SNMP trap logging option and at least one trap listener is configured on the appliance. Now, you can specify the audit log level of trap messages sent to an external log server. The default log level is Informational. Possible values are Emergency, Alert, Critical, Error, Warning, Debug, and Notice.

For example, you can set the audit log level to Critical for an SNMP trap message generated by a logon failure. That information is then available on the NSLOG or SYSLOG server for troubleshooting.

To enable SNMP trap logging and configure trap log level by using the command line interface

At the command prompt, type the following commands to configure SNMP trap logging and verify the configuration:

- `set snmp option [-snmpTrapLogging (ENABLED | DISABLED)][-snmpTrapLoggingLevel <snmpTrapLoggingLevel>]`
- `show snmp option`

To enable SNMP trap logging and configure SNMP trap log level by using the GUI

Navigate to **System** > **SNMP**, click Change SNMP Options and set the following parameters:

1. SNMP Trap Logging—Select this check box to enable SNMP trap logging when at least one trap listener is configured on the appliance.
2. SNMP Trap Logging Level—Select an audit log level for the SNMP trap. By default, the audit level for SNMP trap is set to "Informational."

Configuring NetScaler for SNMP v1 and v2 queries

September 21, 2020

You can query the NetScaler SNMP agent for system-specific information from a remote device called *SNMP managers*. The agent then searches the management information base (MIB) for the data requested and sends the data to the SNMP manager.

The following types of SNMP v1 and v2 queries are supported by the SNMP agent:

- GET
- GET NEXT
- ALL
- GET BULK

You can create strings called community strings and associate each of these to query types. You can associate one or more community strings to each query type. Community string are passwords and used to authenticate SNMP queries from SNMP managers.

For example, if you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the NetScaler appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

Specifying an SNMP manager

You must configure the NetScaler appliance to allow the appropriate SNMP managers to query it. You must also provide the SNMP manager with the required NetScaler-specific information. You can add up to a maximum of 100 SNMP managers or networks.

For an IPv4 SNMP manager you can specify a host name instead of the manager's IP address. If you do so, you must add a DNS name server that resolves the host name of the SNMP manager to its IP address. You can add up to a maximum of five host-name based SNMP managers.

Note: The appliance does not support use of host names for SNMP managers that have IPv6 addresses. You must specify the IPv6 address.

If you do not configure at least one SNMP manager, the appliance accepts and responds to SNMP queries from all IP addresses on the network. If you configure one or more SNMP managers, the appliance accepts and responds only to SNMP queries from those specific IP addresses.

If you remove an SNMP manager from the configuration, that manager can no longer query the appliance.

To add SNMP managers by specifying IP addresses by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp manager <IPAddress> ... [-netmask <netmask>]`
- `show snmp manager`

Example

```
> add snmp manager 10.102.29.10 10.102.29.15 10.102.29.30
```

To add an SNMP manager by specifying its host name by using the command line interface

Important: If you specify the SNMP manager's host name instead of its IP address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see "Adding a Name Server."

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp manager <IPAddress> [-domainResolveRetry ****<integer>]`
- `show snmp manager`

Example

```
add nameserver 10.103.128.15
add snmp manager engwiki.eng.example.net -domainResolveRetry 10
```

To add an SNMP manager by using the GUI

1. Navigate to **System** > **SNMP** > **Managers**, and create the SNMP manager.

   > **Important**:
   >
   > If you specify the SNMP manager's host name instead of its IPv4 address, you must configure a DNS name server to resolve the host name to the SNMP manager's IP address. For more information, see "Adding a Name Server."

> **Note**:
> The appliance does not support host names for SNMP managers that have IPv6 addresses.

Specifying an SNMP community

You can create strings called community strings and associate them with the following SNMP query types on the appliance:

- GET
- GET NEXT
- ALL
- GET BULK

You can associate one or more community strings to each query types. For example, when you associate two community strings, such as **abc** and **bcd**, to the query type GET NEXT, the SNMP agent on the appliance considers only those GET NEXT SNMP query packets that contain **abc** or **bcd** as the community string.

If you do not associate any community string to a query type then the SNMP agent responds to all SNMP queries of that type.

To specify an SNMP community by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp community <communityName> <permissions>`
- `show snmp community`

Example

```
> add snmp community com all
```

To configure an SNMP community string by using the GUI

Navigate to **System** > **SNMP** > **Community**, and create the SNMP community.

Configuring NetScaler for SNMPv3 queries

September 21, 2020

Simple Network Management Protocol Version 3 (SNMPv3) is based on the basic structure and architecture of SNMPv1 and SNMPv2. However, SNMPv3 enhances the basic architecture to incorporate administration and security capabilities, such as authentication, access control, data integrity check, data origin verification, message timeliness check, and data confidentiality.

To implement message level security and access control, SNMPv3 introduces the user-based security model (USM) and the view-based access control model (VACM).

- **User-Based Security Model.** The user-based security model (USM) provides message-level security. It enables you to configure users and security parameters for the SNMP agent and the SNMP manager. USM offers the following features:
  - **Data integrity:** To protect messages from being modified during transmission through the network.
  - **Data origin verification:** To authenticate the user who sent the message request.
  - **Message timeliness:** To protect against message delays or replays.
  - **Data confidentiality:** To protect the content of messages from being disclosed to unauthorized entities or individuals.
- **View-Based Access Control Model.** The view-based access control model (VACM) enables you to configure access rights to a specific subtree of the MIB based on various parameters, such as security level, security model, user name, and view type. It enables you to configure agents to provide different levels of access to the MIB to different managers.

NetScaler supports the following entities that enable you to implement the security features of SNMPv3:

- SNMP Engines
- SNMP Views
- SNMP Groups
- SNMP Users

These entities function together to implement the SNMPv3 security features. Views are created to allow access to subtrees of the MIB. Then, groups are created with the required security level and access to the defined views. Finally, users are created and assigned to the groups.

> **Note**:
>
> The view, group, and user configuration are synchronized and propagated to the secondary node in a high availability (HA) pair. However, the engine ID is neither propagated nor synchronized as it is unique to each NetScaler appliance.

To implement message authentication and access control, you need to do the following:

Setting the engine ID

SNMP engines are service providers that reside in the SNMP agent. They provide services such as sending, receiving, and authenticating messages. SNMP engines are uniquely identified using engine IDs.

The NetScaler appliance has a unique engineID based on the MAC address of one of its interfaces. It is not necessary to override the engineID. However, if you want to change the engine ID, you can reset it.

To set the engine ID by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `set snmp engineId <engineID>`
- `show snmp engineId`

Example

```
> set snmp engineId 8000173f0300c095f80c68
```

To set the engine ID by using GUI

Navigate to **System** > **SNMP** > **Users**, click **Configure Engine ID** and type an engine ID.

Configure a view

SNMP views restrict user access to specific portions of the MIB. SNMP views are used to implement access control.

To add an SNMP view by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp view <name> <subtree> -type ( included | excluded )`
- `show snmp view <name>`
- `rn snmp view <name> <subtree>`

Where,

Name. Name for the SNMPv3 view. It can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 view.

Subtree. A particular branch (subtree) of the MIB tree that you want to associate with this SNMPv3 view. You must specify the subtree as an SNMP OID. This is an argument of maximum Length: 99.

type. Include or exclude the subtree, specified by the subtree parameter, in or from this view. This setting can be useful when you have included a subtree, such as A, in an SNMPv3 view and you want to exclude a specific subtree of A, such as B, from the SNMPv3 view. This is a mandatory argument. Possible values: included, excluded.

Examples

add snmp view SNMPv3test 1.1.1.1 -type included
sh snmp view SNMPv3test
rm snmp view SNMPv3test 1.1.1.1

To configure an SNMP view by using the GUI

Navigate to **System** > **SNMP** > **Views**,and create the SNMP view.

Configure a group

SNMP groups are logical aggregations of SNMP users. They are used to implement access control and to define the security levels. You can configure an SNMP group to set access rights for users assigned to that group, thereby restricting the users to specific views.

You need to configure an SNMP group to set access rights for users assigned to that group.

To add an SNMP group by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp group <name> <securityLevel> -readViewName <string>`
- `show snmp group <name> <securityLevel>`

Where,

Name. Name for the SNMPv3 group. Can consist of 1 to 31 characters that include uppercase and lowercase letters, numbers, and the hyphen (-), period (.) pound (#), space ( ), at sign (@), equals (=), colon (:), and underscore (_) characters. You should choose a name that helps identify the SNMPv3 group.

securityLevel. Security level required for communication between the NetScaler appliance and the SNMPv3 users who belong to the group. Specify one of the following options:

noAuthNoPriv. Require neither authentication nor encryption.

authNoPriv. Require authentication but no encryption.

authPriv. Require authentication and encryption. Note: If you specify authentication, you must specify an encryption algorithm when you assign an SNMPv3 user to the group. If you also specify encryption, you must assign both an authentication and an encryption algorithm for each group member. This is a mandatory argument. Possible values: noAuthNoPriv, authNoPriv, authPriv.

readViewName. Name of the configured SNMPv3 view that you want to bind to this SNMPv3 group. An SNMPv3 user bound to this group can access the subtrees that are bound to this SNMPv3 view as type INCLUDED, but cannot access the ones that are type EXCLUDED. If the NetScaler appliance has multiple SNMPv3 view entries with the same name, all such entries are associated with the SNMPv3 group. This is a mandatory argument. Maximum Length: 31

To configure an SNMP group by using the GUI

Navigate to **System** > **SNMP** > **Groups**, and create the SNMP group.

Configuring a user

SNMP users are the SNMP managers that the agents allow to access the MIBs. Each SNMP user is assigned to an SNMP group.

You need to configure users at the agent and assign each user to a group.

To configure a user by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `add snmp user <name> -group <string> [-authType ( MD5 | SHA ){ -authPasswd } [-privType ( DES | AES ){ -privPasswd } ]]`
- `show snmp user <name>`

Where,

authType is the authentication option available while configuring an user. There are two authentication types such as MD5 and SHA.

privType is the encryption option available while configuring an user. There are two types of encryption such as DES of key size 128 bit, and AES of key size 128 bit.

Example

```
> add snmp user edocs_user -group edocs_group
<!--NeedCopy-->
```

To configure an SNMP user by using the GUI

Navigate to **System** > **SNMP** > **Users**, and create the SNMP user.

Configuring SNMP Alarms for rate limiting

September 21, 2020

NetScaler appliances such as the NetScaler MPX 10500, 12500, and 15500 are rate limited. The maximum throughput (Mbps) and packets per second (PPS) are determined by the license purchased for the appliance. For rate-limited platforms, you can configure SNMP traps to send notifications when throughput and PPS approach their limits and when they return to normal.

Throughput and PPS are monitored every seven seconds. You can configure traps with high-threshold and normal-threshold values, which are expressed as a percentage of the licensed limits. The appliance then generates a trap when throughput or PPS exceeds the high threshold, and a second trap when the monitored parameter falls to the normal threshold. In addition to sending the traps to the configured destination device, the NetScaler logs the events associated with the traps in the /var/log/ns.log file as EVENT ALERTSTARTED and EVENT ALERTENDED.

Exceeding the throughput limit can result in packet loss. You can configure SNMP alarms to report packet loss.

For more information about SNMP alarms and traps, see "Configuring the NetScaler to generate SNMP v1 and v2 Traps."

This document includes the following details:

- Configuring an SNMP Alarm for Throughput or PPS
- Configuring SNMP Alarm for Dropped Packets

Configuring an SNMP alarm for throughput or PPS

To monitor both throughout and PPS, you must configure seperate alarms and set threshold pps value in Mbps.

To configure an SNMP alarm for the throughput rate by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm, set threshold value in Mbps and verify the configuration:

- `set snmp alarm PF-RL-RATE-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-RATE-THRESHOLD`

Example

```
> set snmp alarm PF-RL-RATE-THRESHOLD -thresholdValue 70 -normalValue 50
<!--NeedCopy-->
```

To configure an SNMP alarm for PPS by using the command line interface

At the command prompt, type the following commands to configure the SNMP alarm for PPS and verify the configuration:

- `set snmp alarm PF-RL-PPS-THRESHOLD [-thresholdValue <positive_integer> [-normalValue <positive_integer>]] [-state ( ENABLED | DISABLED )] [-severity <severity>] [-logging ( ENABLED | DISABLED )]`
- `show snmp alarm PF-RL-PPS-THRESHOLD`

Example

```
> set snmp alarm PF-RL-PPS-THRESHOLD -thresholdValue 70 -normalValue 50
<!--NeedCopy-->
```

To configure an SNMP alarm for throughput or PPS by using the GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select **PF-RL-RATE-THRESHOLD** (for throughput rate) or **PF-RL-PPS-THRESHOLD** (for packets per second).
2. Set the alarm parameters and enable the selected SNMP alarm.

Configuring SNMP alarm for dropped packets

You can configure an alarm for packets dropped as a result of exceeding the throughput limit and an alarm for packets dropped as a result of exceeding the PPS limit.

To configure an SNMP alarm for packets dropped because of excessive throughput, by using the command line interface

At the command prompt, type:

```
set snmp alarm PF-RL-RATE-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure an SNMP alarm for packets dropped because of excessive PPS, by using the command line interface

At the command prompt, type:

```
set snmp alarm PF-RL-PPS-PKTS-DROPPED [-state (ENABLED | DISABLED)] [-
severity <severity>] [-logging ( ENABLED | DISABLED )]
```

To configure an SNMP alarm for dropped packets by using the GUI

1. Navigate to **System** > **SNMP** > **Alarms**, and select **PF-RL-RATE-PKTS-DROPPED** (for packets dropped because of excessive throughput) or **PF-RL-PPS-PKTS-DROPPED** (for packets dropped because of excessive PPS).
2. Set the alarm parameters and enable the selected SNMP alarm.

Audit logging

September 21, 2020

Auditing is a methodical examination or review of a condition or situation. The Audit Logging feature enables you to log the NetScaler states and status information collected by various modules in the kernel and in the user-level daemons. For audit logging, you can use the SYSLOG protocol, the native NSLOG protocol, or both.

SYSLOG is a standard protocol for logging. It has two components: the SYSLOG auditing module, which runs on the NetScaler appliance, and the SYSLOG server, which can run on the underlying FreeBSD operating system (OS) of the NetScaler appliance or on a remote system. SYSLOG uses user data protocol (UDP) for data transfer.

Similarly, the native NSLOG protocol has two components the NSLOG auditing module, which runs on the NetScaler appliance, and the NSLOG server, which can run on the underlying FreeBSD OS of the NetScaler appliance or on a remote system. NSLOG uses transmission control protocol (TCP) for data transfer.

When you run a SYSLOG or NSLOG server, it connects to the NetScaler appliance. The NetScaler appliance then starts sending all the log information to the SYSLOG or NSLOG server, and the server can filter the log entries before storing them in a log file. An NSLOG or SYSLOG server can receive log information from more than one NetScaler appliance, and a NetScaler appliance can send log information to more than one SYSLOG server or NSLOG server.

If multiple SYSLOG servers are configured, the NetScaler appliance send its SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastPackets, and AuditlogHash.

> **Note**
>
> The NetScaler appliance can send audit log messages up to 16 KB to an external SYSLOG server.

The log information that a SYSLOG or NSLOG server collects from a NetScaler appliance is stored in a log file in the form of messages. These messages typically contain the following information:

- The IP address of a NetScaler appliance that generated the log message.
- A time stamp
- The message type
- The predefined log levels (Critical, Error, Notice, Warning, Informational, Debug, Alert, and Emergency)
- The message information

To configure audit logging, you first configure the audit modules on the NetScaler appliance. That involves creating audit policies and specifying the NSLOG server or SYSLOG server information. You then install and configure the SYSLOG or the NSLOG server on the underlying FreeBSD OS of the NetScaler appliance or on a remote system.

> **Note**:
>
> Because SYSLOG is an industry standard for logging program messages, and various vendors provide support, this documentation does not include SYSLOG server configuration information.

The NSLOG server has its own configuration file (auditlog.conf). You can customize logging on the NSLOG server system by making additional modifications to the configuration file (auditlog.conf).

Configuring NetScaler appliance for audit logging

October 14, 2020

Auditing module allows logging of all states and status information from different modules so that an administrator can see event history in the chronological order. Main components of Audit framework are ‘audit action’, ‘audit policy’. ‘Audit action’ describes Audit Server configuration information whereas ‘audit policy’ links a bind entity to an ‘audit action’. The audit policies use ‘Classic Policy Engine’(CPE) framework to link ‘audit action’ to ‘bind entities’ or Progress Integration (PI) framework to link ‘audit action’ to ‘system global bind entities’.

The policy frameworks differ from each other in the way the audit-log policies are bound to the global entities. Previously, the audit module supported only Classic expression but now it supports both Classic and Advanced policy expressions. Currently, the Advanced expression can bind audit-log policies only to System global entities.

> **Note**
>
> When binding a policy to global entities, you must bind a policy it to a system global entity of the same expression type. For example, you cannot bind a classic policy to an advanced global entity or bind an advanced policy to a classic global entity.

Configuring audit-log policies in a Classic policy expression

Configuring audit-logging in Classic policy consists of the following steps:

1. **Configuring an audit-log action**. You can configure an audit action for different servers and for different log levels. ‘Audit action’ describes Audit Server configuration information whereas ‘audit policy’ links a bind entity to an ‘audit action’. By default, the SYSLOG, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data. When using TCP for SYSLOG, you can set the buffer limit on the NetScaler appliance to store the logs. After the after which the logs are sent to the SYSLOG server.
2. **Configuring audit-log policy**. Configure SYSLOG policies to log messages to a SYSLOG server, and NSLOG policy to log messages to an NSLOG server. Each policy includes a rule identifying the messages to be logged, and a SYSLOG or NS LOG action.
3. **Binding audit-log policies to global entities**. You must globally bind the audit log policies to global entities such SYSTEM, VPN, AAA, and so on, to enable logging of all NetScaler system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

Each of these steps is explained in the following sections.

Configuring audit-log action

To configure SYSLOG action in the Advanced Policy infrastructure by using the command line interface.

> **Note**
>
> The NetScaler appliance allows you to configure only one SYSLOG action to SYSLOG server IP address and port. The appliance does not allow you to configure multiple SYSLOG actions to the same server IP address and port.

A syslog action contains a reference to a syslog server. It specifies which information to log and mentions how to log that information.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  add audit syslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )] [-transport ( TCP | UDP )]`
-  show audit syslogAction [<name>]

<!--NeedCopy-->
```

To configure NSLOG action in the Advanced Policy infrastructure by using the command line interface

A nslog action contains a reference to a nslog server. It specifies which information to log and mentions how to log that information.

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  add audit nslogAction <name> <serverIP> [-serverPort <port>] -logLevel <logLevel> [-dateFormat ( MMDDYYYY | DDMMYYYY )]
-  show audit nslogAction [<name>]
<!--NeedCopy-->
```

Configuring audit-log policies

To configure audit-log Policies in Classic Policy infrastructure by using the command line interface

At the command prompt, type:

```
-  add audit syslogpolicy <name> <-rule> <action>
-  add audit nslogpolicy <name> < rule> <action>rm audit nslogpolicy <name>show audit nslogpolicy [<name>]set audit nslogpolicy <name>  [-rule <expression>] [-action <name>]
<!--NeedCopy-->
```

Binding audit-log policies to global entities

To bind audit-log policy in the Classic policy framework by using the command line interface

At the command prompt, type:

`bind auditlog systemglobal <auditlog policy> -globalType SYSTEM -priority`

Configuring audit-log policies using advanced policy expression

Configuring audit-logging in Advanced policy consists of the following steps:

1. **Configuring an audit-log action**. You can configure an audit action for different servers and for different log levels. 'Audit action' describes Audit Server configuration information whereas 'audit policy' links a bind entity to an 'audit action'. By default, the SYSLOG, and NSLOG uses only TCP to transfer log information to the log servers. TCP is more reliable than UDP for transferring complete data. When using TCP for SYSLOG, you can set the buffer limit on the NetScaler appliance to store the logs. After the after which the logs are sent to the SYSLOG server.
2. **Configuring audit-log policy**. Configure SYSLOG policies to log messages to a SYSLOG server, and NSLOG policy to log messages to an NSLOG server. Each policy includes a rule identifying the messages to be logged, and a SYSLOG or NS LOG action.
3. **Binding audit-log policies to global entities**. You must globally bind the audit log policies to SYSTEM global entity to enable logging of all NetScaler system events. By defining the priority level, you can set the evaluation order of the audit server logging. Priority 0 is the highest and is evaluated first. The higher the priority number, the lower is the priority of evaluation.

Configuring audit-log action

To configure syslog action in Advanced Policy infrastructure by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-domainResolveRetry <integer>])
    | -lbVserverName <string>))[-serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat>]
     [-logFacility <logFacility>][-tcp ( NONE | ALL )] [-acl ( ENABLED | DISABLED )]
    [-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES | NO )]
     [-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )]
    [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )][-tcpProfileName <string>][-maxLogDataSizeToHold]
```

```
-  show audit syslogAction [<name>]
<!--NeedCopy-->
```

To configure NSLOG action in Advanced Policy infrastructure by using the command line interface

At the command prompt, type the following commands to set the parameters and verify the configuration:

```
-  add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-
   domainResolveRetry <integer>])) [-serverPort <port>]       -logLevel
    <logLevel> ... [-dateFormat <dateFormat>][-logFacility <logFacility
   >] [-tcp ( NONE | ALL )][-acl ( ENABLED | DISABLED )] [-timeZone (
   GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES | NO )][-
   appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED )
   ][-alg ( ENABLED | DISABLED )] [-subscriberLog ( ENABLED | DISABLED
   )]
-  show audit nslogAction [<name>]
<!--NeedCopy-->
```

Binding audit-log policies to global entities

To bind syslog audit-log policy in Advanced policy framework by using the command line interface

At the command prompt, type:

```
bind audit syslogGlobal <policyName> [-globalBindType <globalBindType
```

```
unbind audit syslogGlobal <policyName>[-globalBindType <globalBindType>]
```

Configuring audit-log policy by using the GUI

1. Navigate to **Configuration** > **System** > **Auditing** > **Syslog** and then to **Policies** tab page to bind a syslog policy (advanced or classic) to system global entity. **Note**: System global can be either in "Classic" or "Advanced" mode. In "Classic" mode, you can bind only a classic audit-log policy and when in "Advanced" mode, you can bind only an advanced audit-log policy.
2. Select a policy and click **Action** to select a system global binding (Advanced or Classic) from the drop-down list.
3. In the **Auditing Syslog Classic Policy Global Binding** page, select a syslog policy and click **Add Binding** to bind the policy to system global entities (in Classic or Advance mode).

```
    > add audit syslogaction audit-action1 10.102.1.1 -loglevel
      INFORMATIONAL -dateformat MMDDYYYY
    > add audit nslogAction nslog-action1 10.102.1.3 -serverport 520 -
      loglevel INFORMATIONAL -dateFormat MMDDYYYY
    > add audit syslogpolicy syslog-pol1 ns_true audit-action1
    > add audit nslogPolicy nslog-pol1 ns_true nslog-action1
    > bind system global nslog-pol1 -priority 20
<!--NeedCopy-->
```

Configuring policy-based logging

You can configure policy-based logging for rewrite and responder policies. Audit messages are then logged in a defined format when the rule in a policy evaluates to TRUE. To configure policy-based logging, you configure an audit-message action that uses default syntax expressions to specify the format of the audit messages, and associate the action with a policy. The policy can be bound either globally or to a load balancing or content switching virtual server. You can use audit-message actions to log messages at various log levels, either in syslog format only or in both syslog and newnslog formats

Prerequisites

- User Configurable Log Messages (userDefinedAuditlog) option is enabled for when configuring the audit action server to which you want to send the logs in a defined format.
- The related audit policy is bound to system global.

Configuring an audit message action

You can configure audit message actions to log messages at various log levels, either in syslog format only or in both syslog and newnslog formats. Audit-message actions use expressions to specify the format of the audit messages.

To create an audit message action by using the command line interface

At the command prompt, type:

```
add audit messageaction <name> <logLevel> <stringBuilderExpr> [-
    logtoNewnslog (YES|NO)] [-bypassSafetyCheck (YES|NO)]
<!--NeedCopy-->
```

```
add audit messageaction log-act1 CRITICAL '"Client:"+CLIENT.IP.SRC+"
    accessed "+HTTP.REQ.URL' -bypassSafetyCheck YES
<!--NeedCopy-->
```

To configure an audit message action by using the GUI

Navigate to **System** > **Auditing** > **Message Actions**, and create the audit message action.

Binding audit message action to a policy

After you create an audit message action, you must bind it to a rewrite or responder policy. For more information about binding log message actions to a rewrite or responder policy, see "Rewrite or "Responder.

Installing and configuring the NSLOG server

September 21, 2020

During installation, the NSLOG server executable file (auditserver) is installed along with other files. The auditserver executable file includes options for performing several actions on the NSLOG server, including running and stopping the NSLOG server. In addition, you use the auditserver executable to configure the NSLOG server with the IP addresses of the NetScaler appliances from which the NSLOG server will start collecting logs. Configuration settings are applied in the NSLOG server configuration file (auditlog.conf).

Then, you start the NSLOG server by executing the auditserver executable. The NSLOG server configuration is based on the settings in the configuration file. You can further customize logging on the NSLOG server system by making additional modifications to the NSLOG server configuration file (auditlog.conf).

> **Attention**:
>
> The version of the NSLOG server package must be the same as that of the NetScaler. For example, if the version of the NetScaler is 10.1 Build 125.9, the NSLOG server must also be of the same version.

The following table lists the operating systems on which the NSLOG server is supported.

Operating system	Software requirements	Remarks
Windows	Windows XP Professional, Windows Server 2003, Windows 2000/NT, Windows Server 2008, Windows Server 2008 R2	

Operating system	Software requirements	Remarks
Linux	RedHat Linux 4 or later, SUSE Linux Enterprise 9.3 or later	
FreeBSD	FreeBSD 6.3 or later	For NetScaler 10.5, use only FreeBSD 8.4.
Mac OS	Mac OS 8.6 or later	Not supported on NetScaler 10.1 and later releases.

The minimum hardware specifications for the platform running the NSLOG server are as follows:

- Processor- Intel x86 ~501 megahertz (MHz)
- RAM - 512 megabytes (MB)
- Controller - SCSI

This document includes the following details:

Installing NSLOG server on the Linux operating system

Log on to the Linux system as an administrator. Use the following procedure to install the NSLOG server executable files on the system.

To install the NSLOG server package on a Linux operating system

1. At a Linux command prompt, type the following command to copy the NSauditserver.rpm file to a temporary directory:

   `cp <path_to_cd>/Utilities/auditserver/Linux/NSauditserver.rpm /tmp`

2. Type the following command to install the NSauditserver.rpm file.

   rpm -i NSauditserver.rpm

   This command extracts the files and installs them in the following directories:

   - `/usr/local/netscaler/etc`
   - `/usr/local/netscaler/bin`
   - `/usr/local/netscaler/samples`

To uninstall the NSLOG server package on a Linux operating system

1. At a command prompt, type the following command to uninstall the audit server logging feature:

`rpm -e NSauditserver`

2. For more information about the NSauditserver RPM file, use the following command:

   `rpm -qpi \*.rpm`

3. To view the installed audit server files use the following command:

   `rpm -qpl *.rpm`

   `*.rpm`: Specifies the file name.

Installing NSLOG server on the FreeBSD operating system

Before you can install the NSLOG server, you have to copy the NSLOG package from the NetScaler product CD or download it from www.citrix.com. The NSLOG package has the following name format:

`AuditServer_<release number>-<build number>.zip`

For example: `AuditServer_10.5-58.11.zip`

This package contains files for all supported platforms: Linux, Windows, and FreeBSD. On a FreeBSD operating system, install the NSLOG package that has the following name format:

`audserver_bsd-<release number>-<build number>.tgz`

For example: `audserver_bsd-10.5-58.11.tgz`

To download NSLOG package from www.citrix.com:

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click **Log In**.
3. Enter your login credentials, and then click **Log In**.
4. In the menu bar, click **Downloads**.
5. From the **Select a product** list, select **NetScaler**.
6. On the **NetScaler** page, select the release for which you want to download the NSLOG package (for example, Release 10.5), and then select **Firmware**.
7. Under **Firmware**, select the NetScaler firmware for the build number for which you want to download the NSLOG package.
8. On the page that appears, scroll down, select **Audit Servers**, and click **Download File** next to the package that you want to download.

To install the NSLOG server package on a FreeBSD operating system

1. On the system to which you have downloaded the NSLOG package `AuditServer_<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`), extract the `FreeBSD NSLOG server package audserver_bsd-<release number>-<build number>.tgz` (for example, `audserver_bsd-9.3-51.5.tgz`) from the package.

2. Copy the FreeBSD NSLOG server package `audserver_bsd-<release number>-<build number>.tgz` (for example, `audserver_bsd-9.3-51.5.tgz`) to a directory on a system running FreeBSD OS.
3. At a command prompt for the directory into which the FreeBSD NSLOG server package was copied, run the following command to install the package:

   `pkg_add audserver_bsd-<release number>-<build number>.tgz`

   Example:

   ```
   pkg_add audserver_bsd-9.3-51.5.tgz
   <!--NeedCopy-->
   ```

   The following directories are extracted:

   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>NetScalerbin` (for example, `/var/auditserver/netscaler/bin`)
   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>netscaler/etc` (for example, `/var/auditserver/netscaler/etc`)
   - `<root directory extracted from the FreeBSD NSLOG server package tgz file>\netscaler\samples` (for example, `/var/auditserver/samples`)
4. At a command prompt, type the following command to verify that the package is installed:

   `pkg_info | grep NSaudserver`

To uninstall the NSLOG server package on a FreeBSD operating system

At a command prompt, type:

`pkg_delete NSaudserver`

Installing NSLOG Server Files on the Windows Operating System

Before you can install the NSLOG server, you have to copy the NSLOG package from the NetScaler product CD or download it from www.citrix.com. The NSLOG package has the following name format `AuditServer _<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`). This package contains NSLOG installation packages for all supported platforms.

To download NSLOG package from www.Citrix.com

1. In a web browser, go to www.citrix.com.
2. In the menu bar, click Log In.
3. Enter your login credentials, and then click Log In.

4. In the menu bar, click Downloads.
5. Search to find the page that provides the appropriate release number and build.
6. On that page, under Audit Servers, click Download to download the NSLOG package, having the format `AuditServer_<release number>-<build number>.zip`, to your local system (for example, `AuditServer_9.3-51.5.zip`).

To install NSLOG server on a Windows operating system

1. On the system, where you have downloaded the NSLOG package `AuditServer_<release number>-<build number>.zip` (for example, `AuditServer_9.3-51.5.zip`), extract `audserver_win-<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) from the package.

2. Copy the extracted file `audserver_<release number>-<build number>.zip` (for example, `audserver_win-9.3-51.5.zip`) to a Windows system on which you want to install the NSLOG server.

3. Unzip the `audserver_<release number>-<build number>.zip` file (for example, audserver_win-9.3-51.5.zip‘).

4. The following directories are extracted:

    a) `<root directory extracted from the Windows NSLOG server package zip file>\bin`(for example, `C:\audserver_win-9.3-51.5\bin`)
    b) `<root directory extracted from the Windows NSLOG server package zip file>\etc` (for example, `C:\audserver_win-9.3-51.5\ etc`)
    c) `<root directory extracted from the Windows NSLOG server package zip file \>\samples` (for example, `C:\audserver_win-9.3-51.5\ samples`)

5. At a command prompt, run the following command from the `<root directory extracted from the Windows NSLOG server package zip file>\bin path`

    ```
    audserver -install -f <directorypath>\auditlog.conf
    ```

    `<directorypath>`: Specifies the path to the configuration file (`auditlog.conf`). By default, `log.conf` is under `\<root directory extracted from Windows NSLOG server package zip file\>\samples` directory. But you can copy auditlog.conf to your desired directory.

To uninstall the NSLOG server on a Windows operating system

At a command prompt, run the following from the `<root directory extracted from Windows NSLOG server package zip file>\bin` path:

```
audserver -remove
```

NSLOG Server Command Options

For information about NSLOG server commands, see Audit Server Options topic.

Run the audserver command from the directory in which the audit server executable is present:

- 'On Windows: \ns\bin'
- On Solaris and Linux:`\usr\local\netscaler\bin`

The audit server configuration files are present in the following directories:

- On Windows: `\ns\etc`
- On Linux: `\usr\local\netscaler\etc`

The audit server executable is started as `./auditserver` in Linux and FreeBSD.

Adding the NetScaler Appliance IP Addresses on the NSLOG Server

In the configuration file (
`auditlog.conf`), add the IP addresses of the NetScaler appliances whose events must be logged.

To add the IP addresses of the NetScaler appliance

At a command prompt, type the following command:

```
audserver -addns -f <directorypath\>\auditlog.conf
```

`<directorypath>`: Specifies the path to the configuration file (auditlog.conf).

You are prompted to enter the information for the following parameters:

NSIP: Specifies the IP address of the NetScaler appliance, for example, 10.102.29.1.

Userid: Specifies the user name, for example, nsroot.

Password: Specifies the password, for example, nsroot.

If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of the NetScaler appliance event details, you can delete the NSIPs manually by removing the NSIP statement at the end of the auditlog.conf file. For a high availability (HA) setup, you must add both primary and secondary NetScaler IP addresses to auditlog.conf by using the audserver command. Before adding the IP address, make sure the user name and password exist on the system.

Verifying the NSLOG Server Configuration File

Check the configuration file (`audit log.conf`) for syntax correctness to enable logging to start and function correctly.

To verify configuration, at a command prompt, type the following command:

```
audserver -verify -f <directorypath>\auditlog.conf
```

`<directorypath>: Specifies the path to the configuration file (audit log.conf).`

Running the NSLOG server

September 21, 2020

To start audit server logging

Type the following command at a command prompt:

audserver -start -f `<directorypath>\auditlog.conf`

`<directorypath>`: Specifies the path to the configuration file (audit log.conf).

To stop audit server logging that starts as a background process in FreeBSD or Linux

Type the following command:

```
audserver -stop
```

To stop audit server logging that starts as a service in Windows

Type the following command:

```
audserver -stopservice
```

Customizing logging on the NSLOG server

September 21, 2020

You can customize logging on the NSLOG server by making additional modifications to the NSLOG server configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the server system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information from a NetScaler appliance or a set of NetScaler appliances.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

This document includes the following details:

- Creating Filters
- Specifying Log Properties

Creating filters

You can use the default filter definition located in the configuration file (audit log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

> **Note**:
> For consolidated logging, if a log transaction occurs for which there is no filter definition, the default filter is used (if it is enabled.) The only way you can configure consolidated logging of all the NetScaler appliances is by defining the default filter.

To create a filter

At the command prompt, type the following command in the configuration file (auditlog.conf):

```
filter <filterName> [IP <ip>] [NETMASK <mask>] ON | OFF]
<!--NeedCopy-->
```

filterName: Specify the name of the filter (maximum of 64 alphanumeric characters).

ip: Specify the IP addresses.

mask: Specify the subnet mask to be used on a subnet.

Specify ON to enable the filter to log transactions, or specify OFF to disable the filter. If no argument is specified, the filter is ON.

Examples:

```
filter F1 IP 192.168.100.151 ON
<!--NeedCopy-->
```

To apply the filter F2 to IP addresses 192.250.100.1 to 192.250.100.254:

```
filter F2 IP 192.250.100.0 NETMASK 255.255.255.0 ON
<!--NeedCopy-->
```

filterName is a required parameter if you are defining a filter with other optional parameters, such as IP address, or the combination of IP address and Netmask.

Specifying log properties

Log properties associated with the filter are applied to all the log entries present in the filter. The log property definition starts with the key word BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
    logFilenameFormat ...
    logDirectory ...
    logInterval ...
    logFileSizeLimit ....
END
<!--NeedCopy-->
```

Entries in the definition can include the following:

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:
  - Static: A constant string that specifies the absolute path and the file name.
  - Dynamic: An expression that includes the following format specifiers:
    - Date (%{format}t)
    - creates file name with NSIP

  Example:

  ```
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t.log
  <!--NeedCopy-->
  ```

  This creates the first file name as Exmmddyy.log. New files are named: Exmmddyy.log.0, Exmmddyy.log.1, and so on. In the following example, the new files are crated when the file size reaches 100MB.

  Example:

  ```
  LogInterval size
  LogFileSize 100
  LogFileNameFormat Ex%`{
  `%m%d%y }
  t
  <!--NeedCopy-->
  ```

> **Caution**
>
> The date format %t specified in the LogFilenameFormat parameter overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat parameter.

- **logDirectory** specifies the directory name format of the log file. The name of the file can be either of the following:
  - Static: Is a constant string that specifies the absolute path and file name.
  - Dynamic: Is an expression containing the following format specifiers:
    - Date (%{format}t)
    - creates directory with NSIP

  The directory separator depends on the operating system. In Windows, use the directory separator.

  Example:

```
    LogDirectory dir1\dir2\dir3
<!--NeedCopy-->
```

```
In the other operating systems (Linux, FreeBsd, etc.), use the directory separator.
```

- **LogInterval** specifies the interval at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour. Default value.
  - Daily: A file is created very day at midnight.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly : A file is created on the first day of the month at midnight.
  - None: A file is created only once, when audit server logging starts.
  - Size: A file is created only when the log file size limit is reached.

  Example:

```
    LogInterval Hourly
<!--NeedCopy-->
```

- **LogFileSizeLimit** specifies the maximum size (in MB) of the log file. A new file is created when the limit is reached.

> **Note**
>
> You can override the loginterval property by assigning size as its value.

The default LogFileSizeLimit is 10 MB.

Example:

```
    LogFileSizeLimit 35
<!--NeedCopy-->
```

SYSLOG Over TCP

September 21, 2020

Syslog is a standard for sending event notification messages. These messages can be stored locally or on an external log server. Syslog enables network administrators to consolidate log messages and derive insights from the collected data.

Syslog was originally designed to work over UDP, which can transmit a huge amount of data within the same network with minimal packet loss. However, telco operators prefer to transmit syslog data over TCP, because they need reliable, ordered data transmission between networks (for example, telco tracks user activities), and TCP provides retransmission in the event of network failure.

How Syslog over TCP works

To understand how syslog over TCP works, consider two hypothetical cases:

Sam, a network administrator, wants to log significant events on an external syslog server.

XYZ Telecom, an Internet service provider, has to transmit and store a significant amount of data on syslog servers to comply with government regulations.

In both cases, the log messages must be transmitted over a reliable channel and stored safely on an external syslog server. Unlike UDP, TCP establishes a connection, transmits messages securely, and retransmits (from sender to receiver) any data that is corrupted or lost because of network failure.

The NetScaler appliance sends log messages over UDP to the local syslog daemon, and sends log messages over TCP or UDP to external syslog servers.

SNIP support for Syslog

When the audit-log module generates syslog messages, it uses a NetScaler subnet IP (SNIP) address as the source address for sending the messages to an external syslog server. To configure a SNIP as the source address, you must make it part of the netProfile option and bind the netProfile to the syslog action.

> **Note**
>
> If a netProfile is not bound to a syslog action, the NetScaler IP (NSIP) is used as the source address for transmitting data to the external log server.

Use of a SNIP address is not supported in internal logging.

FQDN Support for audit Log

Previously, the audit-log module was configured with the destination IP address of the external syslog server to which the log messages are sent. Now, the audit-log server uses a Fully Qualified Domain Name (FQDN) instead of the destination IP address. The FQDN configuration resolves the configured domain name of the syslog server to the corresponding destination IP address for sending the log messages from the audit-log module. To resolve the domain name and avoid domain based service issues, the name server must be properly configured.

> **Note**
>
> When configuring a FQDN, server domain name configuration of the same NetScaler appliance in syslog action or nslog action is not supported.

Configuring Syslog over TCP by using Command Line Interface

To configure a NetScaler appliance to send syslog messages over TCP by using the command line interface

At the command prompt, type:

```
    add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-domainResolveRetry <integer>]) | -lbVserverName<string>))[-serverPort <port>] -logLevel <logLevel>[-dateFormat <dateFormat>] [-logFacility <logFacility>]   [-tcp ( NONE | ALL )] [-acl ( ENABLED | DISABLED )][-timeZone ( GMT_TIME | LOCAL_TIME )][-userDefinedAuditlog ( YES | NO )][-appflowExport ( ENABLED | DISABLED )] [-lsn ( ENABLED | DISABLED )][-alg ( ENABLED | DISABLED )] [-subscriberLog ( ENABLED | DISABLED )][-transport ( TCP | UDP )] [-tcpProfileName <string>][-maxLogDataSizeToHold <positive_integer>][-dns ( ENABLED | DISABLED )] [-netProfile <string>]
<!--NeedCopy-->
```

```
    add audit syslogaction audit-action1 10.102.1.1 -loglevel INFORMATIONAL -dateformat MMDDYYYY -transport TCP
<!--NeedCopy-->
```

Adding SNIP IP address to netprofile option by using the command line interface

To add a SNIP IP address to netprofile by using the command line interface

At the command prompt, type:

```
    add netProfile <name> [-td <positive_integer>] [-srcIP <string>][-srcippersistency ( ENABLED | DISABLED )][-overrideLsn ( ENABLED | DISABLED )]add syslogaction <name> <serverIP> - loglevel all - netprofile net1
<!--NeedCopy-->
```

```
    add netprofile net1 - srcip 10.102.147.204`
<!--NeedCopy-->
```

where, srcIP is the SNIP.

Adding netprofile in a syslog action by using the command line interface

To add a netProfile option in a syslog action by using the command line interface

At the command prompt, type:

```
     add audit syslogaction <name> (<serverIP> | -lbVserverName <string>)  -logLevel <logLevel>
    -netProfile <string> …

<!--NeedCopy-->
```

```
    add syslogaction sys_act1 10.102.147.36 - loglevel all - netprofile net1
<!--NeedCopy-->
```

Where -net Profile specifies the name of the configured net profile. The SNIP address is configured as part of the netProfile and this netProfile option is bound to the syslog action.

> **Note**
>
> You must always bind the netProfile option to the SYSLOGUDP or SYSLOGTCP services bound to the SYSLOGUDP or SYSLOGTCP load balancing virtual server when LB vserver name is configured in syslogaction.

Configuring FQDN support by using the command line interface

To add a server domain name to a Syslog action by using the command line interface

At the command prompt, type:

```
add audit syslogAction <name> (<serverIP> | ((<serverDomainName>[-domainResolveRetry <integer>]) | -lbVserverName <string>)) -logLevel <logLevel> ...
    set audit syslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>]-serverDomainName <string>] [-lbVserverName <string>]-domainResolveRetry <integer>] [-domainResolveNow]
<!--NeedCopy-->
```

To add a server domain name to a Nslog action by using the command line interface.

At the command prompt, type:

```
    add audit nslogAction <name> (<serverIP> | (<serverDomainName>[-domainResolveRetry <integer>]))  -logLevel <logLevel> ...
    set audit nslogAction <name> [-serverIP <ip_addr|ipv6_addr|*>][-serverDomainName <string>] [-domainResolveRetry <integer>][-domainResolveNow]
<!--NeedCopy-->
```

Where
serverDomainName. Domain name of the log server. This is mutually exclusive with serverIP/ lbVserverName.

DomainResolveRetry integer. Time (in seconds) that the NetScaler appliance waits, after a DNS resolution fails, before sending the next DNS query to resolve the domain name.

DomainResolveNow. Included if the DNS query has to be sent immediately to resolve the server's domain name.

Configuring Syslog over TCP by using the GUI

To configure the NetScaler appliance to send Syslog messages over TCP by using the GUI

1. Navigate to **System** > **Auditing** > **Syslog** and select the **Servers** tab.
2. Click **Add** and select Transport Type as **TCP**.

Configuring netprofile for SNIP support by using the GUI

To configure netprofile for SNIP support by using the GUI

1. Navigate to **System** > **Auditing** > **Syslog** and select the **Servers** tab.
2. Click **Add** and select a netprofile from the list.

Configuring FQDN by using the GUI

To configure FQDN by using the GUI

1. Navigate to **System** > **Auditing** > **Syslog** and select the **Servers** tab.
2. Click **Add** and select a Server Type and Server Domain Name from the list.

Load balancing SYSLOG servers

September 21, 2020

The NetScaler appliance normally sends SYSLOG events and messages to all the configured external log servers. This results in storing redundant messages and makes monitoring difficult for system administrators. To address this issue, the NetScaler appliance offers load balancing algorithms that can load balance the SYSLOG messages among the external log servers for better maintenance and performance. The supported load balancing algorithms include RoundRobin, LeastBandwidth, CustomLoad, LeastConnection, LeastPackets, and AuditlogHash.

Load balancing of SYSLOG servers using the command line interface

At the command prompt, type:

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   ```
   add service <name>(<IP> | <serverName>)<serviceType (SYSLOGTCP |
   SYSLOGUDP)> <port>
   ```

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGUDP, and load balancing method as AUDITLOGHASH.

   ```
   add lb vserver <name> <serviceType (SYSLOGTCP | SYSLOGUDP)> [-lbMethod
   <AUDITLOGHASH>]
   ```

3. Bing the service to the load balancing virtual server.

   ```
   Bind lb vserver <name> <serviceName>
   ```

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   ```
   add syslogaction <name> <serverIP> [-lbVserverName <string>] [-logLevel
    <logLevel>]
   ```

5. Add a SYSLOG policy by specifying the rule and action.

   ```
   add syslogpolicy <name> <rule> <action>
   ```

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   `bind system global <policyName>`

*Load balancing of SYSLOG servers using the GUI

1. Add a service and specify the service type as SYSLOGTCP or SYSLOGUDP.

   Navigate to **Traffic Management** > **Services**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

2. Add a load balancing virtual server, specify the service type as SYSLOGTCP or SYSLOGTCP, and load balancing method as AUDITLOGHASH.

   Navigate to **Traffic Management** > **Virtual Servers**, click **Add** and select **SYLOGTCP** or **SYSLOGUDP** as protocol.

3. Bing the service to the load balancing virtual server to the service.

   Bing the service to the load balancing virtual server.

   Navigate to **Traffic Management** > **Virtual Servers**, select a virtual server and then select **AUDITLOGHASH** in the **Load Balancing Method.**

4. Add a SYSLOG action and specify the load balancing server name that has SYSLOGTCP or SYSLOGUDP as service type.

   Navigate to **System** > **Auditing**, click **Servers** and add a server by selecting **LB Vserver** option in **Servers**.

5. Add a SYSLOG policy by specifying the rule and action.

   Navigate to **System** > **Syslog**, click **Policies** and add a SYSLOG policy.

6. Bind the SYSLOG policy to the system global for the policy to take effect.

   Navigate to **System** > **Syslog**, select a SYSLOG policy and click **Action**, and then click **Global Bindings** and bind the policy to system global.

Example:

The following configuration specifies load balance of SYSLOG messages among the external log servers using the AUDITLOGHASH as load balancing method. AUDITLOGHASH method load balances the traffic based on input hash value from the audit agents. The agents are the modules which generate auditlog in a NetScaler appliance. For example, if an agent LSN wants to load balance auditlogs based on client IP address, LSN module generates the hash value based on clientIP and passes the hash value to auditlog module. The auditlog module sends the auditlog messages which have same hash value to the external syslog server.

The NetScaler appliance generates SYSLOG events and messages that are load balanced amongst the services, service1, service2, and service 3.

```
add service service1 192.0.2.10 SYSLOGUDP 514
add service service2 192.0.2.11 SYSLOGUDP 514
add service service3 192.0.2.11 SYSLOGUDP 514
add lb vserver lbvserver1 SYSLOGUDP -lbMethod AUDITLOGHASH
bind lb vserver lbvserver1 service1
bind lb vserver lbvserver1 service2
bind lb vserver lbvserver1 service3
add syslogaction sysaction1 -lbVserverName lbvserver1 -logLevel All
add syslogpolicy syspol1 ns_true sysaction1
bind system global syspol1
<!--NeedCopy-->
```

Limitations:

- The NetScaler appliance does not support an external load balancing virtual server load balancing the SYSLOG messages among the log servers.

Default settings for the log properties

September 21, 2020

The following is an example of the default filter with default settings for the log properties:

```
begin default
 logInterval Hourly
 logFileSizeLimit 10
 logFilenameFormat  auditlog%`{
 `%y%m%d }
 t.log
 end default
<!--NeedCopy-->
```

Following are two examples of defining the default filters:

Example 1:

```
Filter f1 IP 192.168.10.1
<!--NeedCopy-->
```

This creates a log file for NSI 192.168.10.1 with the default values of the log in effect.

Example 2:

```
Filter f1 IP 192.168.10.1
```

```
begin f1
    logFilenameFormat logfiles.log
end f1
<!--NeedCopy-->
```

This creates a log file for NSIP 192.168.10.1. Since the log file name format is specified, the default values of the other log properties are in effect.

Sample configuration file (audit.conf)

September 21, 2020

Following is a sample configuration file:

```
##############################
# This is the Auditserver configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##############################
MYIP <NSAuditserverIP>
MYPORT 3023
#   Filter filter_nsip  IP <Specify the NetScaler IP address to filter on > ON
#   begin filter_nsip
#       logInterval         Hourly
#       logFileSizeLimit    10
#       logDirectory         logdir\%A\
#       logFilenameFormat   nsip%\\{
 \\%d%m%Y }
 t.log
#   end filter_nsip
Filter default
begin default
    logInterval      Hourly
    logFileSizeLimit    10
    logFilenameFormat   auditlog%\{
 \%y%m%d }
 t.log
end default
<!--NeedCopy-->
```

Web server logging

September 21, 2020

You can use the Web server logging feature to send logs of HTTP and HTTPS requests to a client system for storage and retrieval. This feature has two components:

- The Web log server, which runs on the NetScaler.
- The NetScaler Web Logging (NSWL) client, which runs on the client system.

When you run the NetScaler Web Logging (NSWL) client:

1. It connects to the NetScaler.
2. The NetScaler buffers the HTTP and HTTPS request log entries before sending them to the client.
3. The client can filter the entries before storing them.

To configure Web server logging, you first enable the Web logging feature on the NetScaler and configure the size of the buffer for temporarily storing the log entries. Then, you install NSWL on the client system. You then add the NetScaler IP address (NSIP) to the NSWL configuration file. You are now ready to start the NSWL client to begin logging. You can customize Web server logging by making additional modifications to the NSWL configuration file (log.conf).

Configuring the NetScaler for web server logging

September 21, 2020

To configure the NetScaler for web server logging you are required to only enable the Web Server Logging feature. Optionally, you can perform the following configurations:

- Modify the size of the buffer (default size is 16 MB) that stores the logged information before it is sent to the NetScaler Web Logging (NSWL) client.
- Specify the custom HTTP headers that you want to export to the NSWL client. You can configure a maximum of two HTTP request and two HTTP response header names.

To configure web server logging by using the command line interface

At the command prompt, perform the following operations:

- Enable the web server logging feature.

  ```
  enable ns feature WL
  ```

- [Optional] Modify the buffer size for storing the logged information.

  `set ns weblogparam -bufferSizeMB <size>`

  > **Note**:
  >
  > To activate your modification, you must disable and then re-enable the Web server logging feature.

- [Optional] Specify the custom HTTP header names that you want to export.

  `set ns weblogparam [-customReqHdrs <string> ...] [-customRspHdrs <string> ...]`

```
    > enable ns feature WL
    Done
    > set ns weblogparam -bufferSizeMB 60
    Done
    > show ns weblogparam
        Web Logging parameters:
        Log buffer size:  60MB
        Custom HTTP request headers: (none)
        Custom HTTP response headers: (none)
    Done
    > set ns weblogparam -customReqHdrs req1 req2 -customRspHdrs res1
       res2
    Done
    > show ns weblogparam
        Web Logging parameters:
        Log buffer size:  60MB
        Custom HTTP request headers: req1, req2
        Custom HTTP response headers: res1, res2
    Done
<!--NeedCopy-->
```

To configure web server logging by using the GUI

1. Navigate to **System** > **Settings** and perform the following operations:
   a) To enable the web server logging feature, click **Change Advanced Features** and select **Web Logging**.
   b) To modify the buffer size, click **Change Global System Settings** and under **Web Logging**, enter the buffer size.
   c) To specify the custom HTTP headers to be exported, click **Change Global System Settings** and under **Web Logging**, specify the header values.

Installing the NetScaler web logging (NSWL) client

September 21, 2020

During installation, the NSWL client executable file (nswl) is installed along with other files. The nswl executable file provides a list of options that you can use. For details, see Configuring the NSWL Client.

> **Attention**:
>
> The version of the NSWL client must be the same as that of the Citrix ADC. For example, if the version of the Citrix ADC is 10.1 Build 125.9, the NSWL client must also be of the same version. Also, the web logging (NSWL) client works on both 32 bit and on 64 bit server machines.

The following table lists the operating systems on which the NSWL client can be installed.

Operating system	Version	Hardware requirements	Remarks
Windows	Windows XP Professional, Windows Server 2003, Windows 2000/NT, Windows Server 2008, Windows Server 2008 R2	Processor - Intel x86 ~501 MHz, RAM - 512 MB, Controller - SCSI	
Mac OS	Mac OS 8.6 or later	-	Not supported on NetScaler 10.1 and later releases.
Linux	RedHat Linux 4 or later, SUSE Linux Enterprise 9.3 or later	Processor - Intel x86 ~501 MHz, RAM - 512 MB, Controller - SCSI	
Solaris	Solaris Sun OS 5.6 or later	Processor - UltraSPARC-IIi 400 MHz, RAM - 512 MB, Controller - SCSI	Not supported on NetScaler 10.5 and later releases.
FreeBSD	FreeBSD 6.3 or later	Processor - Intel x86 ~501 MHz, RAM - 512 MB, Controller - SCSI	For NetScaler 10.5, use only FreeBSD 8.4.
AIX	AIX 6.1	-	Not supported on NetScaler 10.5 and later releases.

If the NSWL client system cannot process the log transaction because of a CPU limitation, the Web log buffer overruns and the logging process reinitiates.

Caution: Reinitiation of logging can result in loss of log transactions.

To temporarily solve a NSWL client system bottleneck caused by a CPU limitation, you can tune the Web server logging buffer size on the NetScaler appliance. To solve the problem, you need a client system that can handle the site's throughput.

Downloading the NSWL client

You can obtain the NSWL client package from either the NetScaler product CD or the Citrix downloads site. Within the package there are separate installation packages for each supported platforms.

To download the NSWL client package from the Citrix site

1. Open the URL `https://www.citrix.com/downloads.html`.
2. Log in to the site using your credentials.
3. Open the page for the required release number and build.
4. In the page, under Weblog Clients, click Download. The package has the name format as follows: `Weblog- <release number>-<build number>.zip`.

Installing the NSWL client on a Solaris system

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_solaris-<release number>-<build number>.tar file` from the package.

2. Copy the extracted file to a Solaris system on which you want to install the NSWL client.

3. Extract the files from the tar file with the following command:

   ```
   tar xvf nswl_solaris-9.3-51.5.tar
   <!--NeedCopy-->
   ```

   A directory NSweblog is created in the temporary directory, and the files are extracted to the NSweblog directory.

4. Install the package with the following command:

   ```
   pkgadd -d
   <!--NeedCopy-->
   ```

The list of available packages appears. In the following example, one NSweblog package is shown:

```
1 NSweblog NetScaler Weblogging (SunOS,sparc) 7.0
<!--NeedCopy-->
```

5. You are prompted to select the packages. Select the package number of the NSweblog to be installed.

   After you select the package number and press Enter, the files are extracted and installed in the following directories:

   - `/usr/local/netscaler/etc`
   - `/usr/local/netscaler/bin`
   - `/usr/local/netscaler/samples`

6. To check whether the NSWL package is installed, execute the following command:

```
pkginfo | grep NSweblog
<!--NeedCopy-->
```

> **Note**: To uninstall the NSWL package, execute the following command:

```
pkgrm NSweblog
<!--NeedCopy-->
```

Installing the NSWL Client on a Linux System

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_linux-<release number>-<build number>.rpm` file from the package.
2. Copy the extracted file to a system, running Linux OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:

```
rpm -i nswl_linux-9.3-51.5.rpm
<!--NeedCopy-->
```

   This command extracts the files and installs them in the following directories.

   - `/usr/local/netscaler/etc`
   - `/usr/local/netscaler/bin`
   - `/usr/local/netscaler/samples`

> **Note:** To uninstall the NSWL package, execute the following command:

```
rpm -e NSweblog
<!--NeedCopy-->
```

> **Note:**
> To get more information about the NSweblog RPM file, execute the following command:

```
    rpm -qpi *.rpm
<!--NeedCopy-->
```

> **Note:**
> To view the installed Web server logging files, execute the following command:

```
    rpm -qpl *.rpm
<!--NeedCopy-->
```

Installing the NSWL client on a FreeBSD system

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_bsd-<release number>-<build number>.tgz` file from the package.
2. Copy the extracted file to a system, running FreeBSD OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:

   ```
   pkg_add nswl_bsd-9.3-51.5.tgz
   <!--NeedCopy-->
   ```

   This command extracts the files and installs them in the following directories.

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

   > **Note:** To uninstall the NSWL package, execute the following command:

   ```
   pkg_delete NSweblog
   <!--NeedCopy-->
   ```

4. To verify that the package is installed, execute the following command:

```
pkg_info | grep NSweblog
<!--NeedCopy-->
```

Installing the NSWL Client on a Mac System

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_macos-<release number>-<build number>.tgz` file from the package.
2. Copy the extracted file to a system, running Mac OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:

   ```
   pkg_add nswl_macos-9.3-51.5.tgz
   <!--NeedCopy-->
   ```

   This command extracts the files and installs them in the following directories:

   - /usr/local/netscaler/etc
   - /usr/local/netscaler/bin
   - /usr/local/netscaler/samples

   > **Note:** To uninstall the NSWL package, execute the following command:

   ```
   pkg_delete NSweblog
   <!--NeedCopy-->
   ```

4. To verify that the package is installed, execute the following command:

   ```
   pkg_info | grep NSweblog
   <!--NeedCopy-->
   ```

Installing the NSWL client on a Windows system

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_win-<release number>-<build number>.zip` file from the package.
2. Copy the extracted file to a Windows system on which you want to install the NSWL client.
3. On the Windows system, unzip the file in a directory (referred as `<NSWL-HOME>`). The following directories are extracted from `/bin`, `/etc` and `/samples`.

4. At the command prompt, run the following command from the `<NSWL-HOME>\bin directory`:

   ```
   nswl -install -f <directorypath>\log.conf
   <!--NeedCopy-->
   ```

   where,

   directorypath refers to the path of the configuration file (log.conf). By default, the file is in the `<NSWL-HOME>` etc directory. However, you can copy the configuration file to any other directory.

   > **Note**: To uninstall the NSWL client, at the command prompt, run the following command from the `<NSWL-HOME>\bin directory`:

   ```
   > nswl -remove
   <!--NeedCopy-->
   ```

Installing the NSWL client on a AIX system

To install the NSWL client, perform the following operations on the system where you downloaded the package.

1. Extract the `nswl_aix-<release number>-<build number>.rpm` file from the package.
2. Copy the extracted file to a system, running AIX OS, on which you want to install the NSWL client.
3. To install the NSWL package, execute the following command:

   ```
   rpm -i nswl_aix-9.3-51.5.rpm
   <!--NeedCopy-->
   ```

   This command extracts the files and installs them in the following directories.

   - `/usr/local/netscaler/etc`
   - `/usr/local/netscaler/`
   - `usr/local/netscaler/samples`

   > **Note:** To uninstall the NSWL package, execute the following command:

   ```
   rpm -e NSweblog
   <!--NeedCopy-->
   ```

   > **Note**: To get more information about the NSweblog RPM file, execute the following command:

   ```
   rpm -qpi *.rpm
   <!--NeedCopy-->
   ```

> **Note:** To view the installed Web server logging files, execute the following command:

```
rpm -qpl *.rpm
<!--NeedCopy-->
```

Configure NSWL client

September 21, 2020

After installing the NSWL client, you can configure the NSWL client using the nswl executable. These configurations are then stored in the NSWL client configuration file (log.conf).

> **Note**:
>
> You can further customize logging on the NSWL client system by making additional modifications to the NSWL configuration file (log.conf). For details, see
> Customizing Logging on the NSWL Client System.

The following table describes the commands that you can use to configure the NSWL client.

NSWL command	Specifies
nswl -help	The available NSWL help options.
nswl -addns -f `<path-to-configuration-file>`	The system that gathers the log transaction data. You are prompted to enter the IP address of the NetScaler appliance. Enter a valid user name and password.
nswl -verify -f `<path-to-configuration-file>`	Check for syntax or semantic errors in the configuration file.
nswl -start -f `<path-to-configuration-file>`	Start the NSWL client based on the settings in the configuration file. Note: For Solaris and Linux: To start Web server logging as a background process, type the ampersand sign (&) at the end of the command.
nswl -stop (Solaris and Linux only)	Stop the NSWL client if it was started as a background process; otherwise, use CTRL+C to stop Web server logging.
nswl -install -f `<path-to-configuration-file>` (Windows only)	Install the NSWL client as a service in Windows.

NSWL command	Specifies
nswl -startservice (Windows only)	Start the NSWL client by using the settings in the configuration file specified in the nswl install option. You can also start NSWL client from Start > Control Panel > Services. Note: The NSWL log files will be created in C:\Windows\SysWOW64.
nswl -stopservice (Windows only)	Stops the NSWL client.
nswl -remove	Remove the NSWL client service from the registry.

Run the following commands from the directory in which the NSWL executable is located:

- `Windows: \ns\bin`
- `Solaris and Linux: \usr\local\netscaler\bin`

The Web server logging configuration files are located in the following directory path:

- `Windows: \ns\etc`
- `Solaris and Linux: \usr\local\netscaler\etc`

The NSWL executable is started as .\nswl in Linux and Solaris.

Add the IP addresses of the NetScaler appliance

In the NSWL client configuration file (log.conf), add the NetScaler IP address (NSIP) from which the NSWL client will start collecting logs.

To add the NSIP address of the NetScaler appliance

1. At the client system command prompt, type:

   ```
   nswl -addns -f < directorypath > \log.conf
   ```

   `< directorypath >: Specifies the path to the configuration file (log.conf).`

2. At the next prompt, enter the following information:

   - **NSIP**: Specify the IP address of the NetScaler appliance.
   - Username and Password: Specify the nsroot user credentials of the NetScaler appliance.

> **Note**:
>
> If you add multiple NetScaler IP addresses (NSIP), and later you do not want to log all of NetScaler system log details, you can delete the NSIPs manually by removing the NSIP statement at the end of the log.conf file. During a failover setup, you must add both primary and secondary NetScaler IP addresses to the log.conf by using the command. Before adding the IP address, make sure the user name and password exist on the NetScaler appliances.

Verify the NSWL configuration file

To make sure that logging works correctly, check the NSWL configuration file (log.conf) on the client system for syntax errors.

To verify the configuration in the NSWL configuration file

At the client system command prompt, type:

```
nswl -verify -f <directorypath>\log.conf
```

`< directorypath>`: Specifies the path to the configuration file (log.conf).

Running the NSWL Client

To start Web server logging

At the client system command prompt, type:

```
nswl -start -f <directorypath>\log.conf
```

`<directorypath>`: Specifies the path to the configuration file (log.conf).

To stop Web server logging started as a background process on the Solaris or Linux operating systems

At the command prompt, type:

```
nswl -stop
```

To stop Web server logging started as a service on the Windows operating system

At the command prompt, type:

```
nswl -stopservice
```

Customize logging on the NSWL client system

September 21, 2020

You can customize logging on the NSWL client system by making additional modifications to the NSWL client configuration file (log.conf). Use a text editor to modify the log.conf configuration file on the client system.

To customize logging, use the configuration file to define filters and log properties.

- **Log filters.** Filter log information based on the host IP address, domain name, and host name of the Web servers.
- **Log properties.** Each filter has an associated set of log properties. Log properties define how to store the filtered log information.

Sample configuration file

Following is a sample configuration file:

```
##########
# This is the NSWL configuration file
# Only the default filter is active
# Remove leading # to activate other filters
##########
##########
# Default filter (default on)
# W3C Format logging, new file is created every hour or on reaching 10 MB file size,
# and the file name is Exyymmdd.log
##########
Filter default
begin default
        logFormat               W3C
        logInterval             Hourly
        logFileSizeLimit        10
        logFilenameFormat       Ex%`{
 `%y%m%d }
 t.log
end default
##########
# NetScaler caches example
# CACHE_F filter covers all the transaction with HOST name www.netscaler.com and the listed server ip's
##########
```

```
#Filter CACHE_F HOST www.netscaler.com IP 192.168.100.89 192.168.100.95 192.168.100.52 192.168.100.53 ON
##########
# netscaler origin server example
# Not interested in Origin server to Cache traffic transaction logging
##########
#Filter ORIGIN_SERVERS IP 192.168.100.64 192.168.100.65 192.168.100.66 192.168.100.67 192.168.100.225 192.168.100.226 192.168.
100.227 192.168.100.228 OFF
##########
# netscaler image server example
# all the image server logging.
##########
#Filter IMAGE_SERVER HOST www.netscaler.images.com IP 192.168.100.71 192.168.100.72 192.168.100.169 192.168.100.170 192.168.10
0.171 ON
##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log.
# Exclude objects that ends with .png .jpg .jar.
##########
#begin ORIGIN_SERVERS
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        40
#       logFilenameFormat       /datadisk5/ORGIN/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logExclude              .png .jpg .jar
#end ORIGIN_SERVERS

##########
# NCSA Format logging, new file is created every day midnight or on reaching 20MB file size,
# and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log with log record timestamp as GMT.
##########
#begin CACHE_F
#       logFormat               NCSA
#       logInterval             Daily
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log
#       logtime                 GMT
#end CACHE_F

##########
# W3C Format logging, new file on reaching 20MB and the log file path name is
# atadisk6/netscaler/log/server's ip/Exmmyydd.log with log record timestamp as LOCAL.
##########
#begin IMAGE_SERVER
#       logFormat               W3C
#       logInterval             Size
#       logFileSizeLimit        20
#       logFilenameFormat /datadisk6/netscaler/log/%AEx%`{
 `%m%d%y }
 t
#       logtime                 LOCAL
#end IMAGE_SERVER

##########
# Virtual Host by Name firm, can filter out the logging based on the host name by,
##########

#Filter VHOST_F IP 10.101.2.151 NETMASK 255.255.255.0
#begin VHOST_F
#       logFormat               W3C
#       logInterval             Daily
#       logFileSizeLimit        10
logFilenameFormat /ns/prod/vhost/%v/Ex%`{
 `%m%d%y }
 t
#end VHOST_F

########## END FILTER CONFIGURATION ##########
<!--NeedCopy-->
```

Creating filters

You can use the default filter definition located in the configuration file (log.conf), or you can modify the filter or create a new filter. You can create more than one log filter.

Note: Consolidated logging, which logs transactions for which no filter is defined, uses the default filter if it is enabled. Consolidated logging of all servers can be done by defining only the default filter.

If the server hosts multiple Web sites and each Web site has its own domain name, and each domain is associated with a virtual server, you can configure Web server logging to create a separate log directory for each Web site. The following table displays the parameters for creating a filter.

Parameter	Specifies	
filterName	Name of the filter. The filter name can include alphanumeric characters and cannot be longer than 59 characters. Filter names longer than 59 characters are truncated to 59 characters.	
HOST name	Host name of the server for which the transactions are being logged.	
IP ip	IP address of the server for which transactions are to be logged (for example, if the server has multiple domains that have one IP address).	
IP ip 2…ip n:	Multiple IP addresses (for example, if the server domain has multiple IP addresses).	
ip6 ip	IPv6 address of the server for which transactions are to be logged.	
IP ip NETMASK mask	IP addresses and netmask combination to be used on a subnet.	
ON	OFF	Enable or disable the filter to log transactions. If no argument is selected, the filter is enabled (ON).

Table 1. Parameters for Creating a Filter

To create a filter

To create a filter, enter the following command in the log.conf file:

- `filter <filterName> <HOST name> | [IP<ip> ] | [IP<ip 2...ip n> ] | <IP ip NETMASK mask> [ON | OFF]`
- `filter <filterName> <HOST name> | [IP6 ip/<prefix length>] [ON | OFF]`

To create a filter for a virtual server

To create a filter for a virtual server, enter the following command in the log.conf file:

`filter <filterName> <VirtualServer IP address>`

Example

In the following example, you specify an IP address of 192.168.100.0 and netmask of 255.255.255.0. The filter applies to IP addresses 192.168.100.1 through 192.168.100.254.

```
Filter F1 HOST www.netscaler.com ON
Filter F2 HOST www.netscaler.com IP 192.168.100.151 ON
Filter F3 HOST www.netscaler.com IP 192.168.100.151 192.165.100.152 ON
Filter F4 IP 192.168.100.151
Filter F5 IP 192.168.100.151 HOST www.netscaler.com OFF
Filter F6 HOST www.netscaler.com HOST www.xyz.com HOST www.abcxyz.com
    IP 192.168.100.200 ON
Filter F7 IP 192.250.100.0 NETMASK 255.255.255.0
Filter F8 HOST www.xyz.com IP 192.250.100.0 NETMASK 255.255.255.0 OFF
For creating filters for servers having IPv6 addresses.
Filter F9 2002::8/112 ON
Filter F10 HOST www.abcd.com IP6 2002::8 ON

<!--NeedCopy-->
```

Specifying log properties

Log properties are applied to all log entries associated with the filter. The log property definition begins with the keyword BEGIN and ends with END as illustrated in the following example:

```
BEGIN <filtername>
 logFormat ...
 logFilenameFormat ...
 logInterval ...
 logFileSize ....
 logExclude ....
 logTime ….
 END
<!--NeedCopy-->
```

Entries in the definition can include the following:

- **LogFormat** specifies the Web server logging feature that supports NCSA, W3C Extended, and custom log file formats.

  By default, the logformat property is w3c. To override, enter custom or NCSA in the configuration file, for example:

```
LogFormat NCSA
<!--NeedCopy-->
```

> **Note**: For the NCSA and custom log formats, local time is used to time stamp transactions and for file rotation.

- **LogInterval** specifies the intervals at which new log files are created. Use one of the following values:
  - Hourly: A file is created every hour.
  - Daily: A file is created every day at midnight. Default value.
  - Weekly: A file is created every Sunday at midnight.
  - Monthly: A file is created on the first day of the month at midnight.
  - None: A file is created only once, when Web server logging starts.

Example:

```
LogInterval Daily
<!--NeedCopy-->
```

- *LogFileSizeLimit** specifies the maximum size of the log file in MB. It can be used with any log interval (weekly, monthly, and so on.) A file is created when the maximum file size limit is reached or when the defined log interval time elapses.

To override this behavior, specify the size as the loginterval property so that a file is created only when the log file size limit is reached.

The default LogFileSizeLimit is 10 MB.

Example:

```
LogFileSizeLimit 35
<!--NeedCopy-->
```

- **LogFilenameFormat** specifies the file name format of the log file. The name of the file can be of the following types:

  - Static: Specifies a constant string that contains the absolute path and file name.
  - Dynamic: Specifies an expression containing the following format:
    - Server IP address
    - Date (%{format}t)
    - URL suffix (%x)
    - Host name (%v)

Example:

```
LogFileNameFormat Ex%`{
 `%m%d%y }
 t.log
<!--NeedCopy-->
```

This command creates the first file name as Exmmddyy.log, then every hour creates a file with file name: Exmmddyy.log.0, Exmmddyy.log.1,…, Exmmddyy.log.n.

Example:

```
LogInterval size
LogFileSize 100
LogFileNameFormat Ex%`{
 `%m%d%y }
 t
<!--NeedCopy-->
```

Caution: The date format %t specified in the LogFilenameFormat command overrides the log interval property for that filter. To prevent a new file being created every day instead of when the specified log file size is reached, do not use %t in the LogFilenameFormat.

- **LogExclude** prevents logging of transactions with the specified file extensions.

Example:

```
LogExclude .html
<!--NeedCopy-->
```

This command creates a log file that excludes log transactions for *.html files.

LogTime specifies log time as either GMT or LOCAL. The defaults are:

- NCSA log file format: LOCAL
- W3C log file format: GMT.

Understanding NCSA and W3C log formats

The NetScaler supports the following standard log file formats:

- NCSA Common Log Format
- W3C Extended Log Format

NCSA Common Log Format

If the log file format is NCSA, the log file displays log information in the following format:

```
Client_IP_address -User_Name [Date:Time -TimeZone] "Method Object
    HTTP_version" HTTP_StatusCode BytesSent
<!--NeedCopy-->
```

To use the NCSA Common log format, enter NCSA in the LogFormat argument in the log.conf file.

The following table describes the NCSA Common log format.

Argument	Specifies
Client_IP_address	The IP address of the client computer.
User Name	The user name.
Date	The date of the transaction.
Time	The time when the transaction was completed.
Time Zone	The time zone (Greenwich Mean Time or local time).
Method	The request method (for example; GET, POST).
Object	The URL.
HTTP_version	The version of HTTP used by the client.
HTTP_StatusCode	The status code in the response.
Bytes Sent	The number of bytes sent from the server.

W3C extended log format

An extended log file contains a sequence of lines containing ASCII characters terminated by either a Line Feed (LF) or the sequence Carriage Return Line Feed (CRLF.) Log file generators must follow the line termination convention for the platform on which they are run.

Log analyzers must accept either LF or CRLF form. Each line may contain either a directive or an entry. If you want to use the W3C Extended log format, enter W3C as the Log-Format argument in the log.conf file.

By default, the standard W3C log format is defined internally as the custom log format, shown as follows:

```
%`{
 `%Y-%m-%d%H:%M:%S }
 t %a %u %S %A %p %m %U %q %s %j %J %T %H %+{
 user-agent }
 i %+{
 cookie }
  i%+{
 referer }
 i
<!--NeedCopy-->
```

You can also change the order or remove some fields in this W3C log format. For example:

```
logFormat W3C %`{
 `%Y-%m-%d%H:%M:%S }
 t %m %U
<!--NeedCopy-->
```

W3C log entries are created with the following format:

```
#Version: 1.0 #Fields: date time cs-method cs-uri #Date: 12-Jun-2001 12:34 2001-06-12 12:34:23 GET /sports/football.html 2001-06-12 12:34:30 GET /sports/football.html
<!--NeedCopy-->
```

Entries

Entries consist of a sequence of fields relating to a single HTTP transaction. Fields are separated by white space; Citrix recommends the use of tab characters. If a field in a particular entry is not used, a dash (-) marks the omitted field.

Directives

Refer to the Directives table for information about the logging process. Lines beginning with the pound sign (#) contain directives.

Example:

The following sample log file shows the log entries in W3C Extended log format:

```
#Version: 1.0 #Fields: time cs-method cs-uri #Date: 12-Jan-1996 00:00:00 00:34:23 GET /sports/football.html 12:21:16 GET /sports/
```

```
    football.html 12:45:52 GET /sports/football.html 12:57:34 GET /
    sports/football.html
<!--NeedCopy-->
```

Fields

The Fields directive lists a sequence of field identifiers that specify the information recorded in each entry. Field identifiers may have one of the following forms:

- **identifier:** Relates to the transaction as a whole.
- **prefix-identifier:** Relates to information transfer between parties defined by the value prefix.
- **prefix (header):** Specifies the value of the HTTP header field header for transfer between parties defined by the value prefix. Fields specified in this manner always have the type.

The following table describes defined prefixes.

Prefix	Specifies
c	Client
s	Server
r	Remote
cs	Client to server
sc	Server to client
sr	Server to remote server (prefix used by proxies)
rs	Remote server to server (prefix used by proxies)
x	Application-specific identifier

Examples:

The following examples are defined identifiers that use prefixes:

cs-method: The method in the request sent by the client to the server.

sc(Referer): The Referer field in the reply.

c-ip: The IP address of the client.

Identifiers

The following table describes the W3C Extended log format identifiers that do not require a prefix.

Identifier	Description
date	The date on which the transaction was done.
time	The time when the transaction is done.

Identifier	Description
time-taken	The time taken (in seconds) for the transaction to complete.
bytes	The number of bytes transferred.
cached	Records whether a cache hit has occurred. A zero indicates a cache miss.

Table 5. W3C Extended Log Format Identifiers (No Prefix Required)

The following table describes the W3C Extended log format identifiers that require a prefix.

Identifier	Description
IP	The IP address and the port number.
dns	The DNS name.
status	The status code.
comment	The comment returned with status code.
method	The method.
url	The URL.
url-stem	The stem portion of the URL.
url-query	The query portion of the URL.

Table 6. W3C Extended Log Format Identifiers (Requires a Prefix)

The W3C Extended Log file format allows you to choose log fields. These fields are shown in the following table.

Field	Description
Date	The date on which the transaction is done.
Time	The time when the transaction is done.
Client IP	The IP address of the client.
User Name	The user name.
Service Name	The service name, which is always HTTP.
Server IP	The server IP address.

Field	Description
Server Port	The server port number
Method	The request method (for example; GET, POST).
Url Stem	The URL stem.
Url Query	The query portion of the URL.
Http Status	The status code in the response.
Bytes Sent	The number of bytes sent to the server (request size, including HTTP headers).
Bytes Received	The number of bytes received from the server (response size, including HTTP headers).
Time Taken	The time taken for transaction to complete, in seconds.
Protocol Version	The version number of HTTP being used by the client.
User Agent	The User-Agent field in the HTTP protocol.
Cookie	The Cookie field of the HTTP protocol.
Referer	The Referer field of the HTTP protocol.

Table 7. W3C Extended Log File Format (Allows Log Fields)

Creating a custom log format

You can customize the display format of the log file data manually or by using the NSWL library. By using the custom log format, you can derive most of the log formats that Apache currently supports.

Creating a custom log format by using the NSWL library

Use one of the following NSWL libraries depending on whether the NSWL executable has been installed on a Windows or Solaris host computer:

- **Windows:** The nswl.lib library located in \ns\bin directory on the system manager host computer.
- **Solaris:** The libnswl.a library located in /usr/local/netscaler/bin.

To create the custom log format by using the NSWL library

1. Add the following two C functions defined by the system in a C source file:

   ns_userDefFieldName() : This function returns the string that must be added as a custom field name in the log record.

   ns_userDefFieldVal() : This function implements the custom field value, then returns it as a string that must be added at the end of the log record.

2. Compile the file into an object file.
3. Link the object file with the NSWL library (and optionally, with third party libraries) to form a new NSWL executable.
4. Add a %d string at the end of the logFormat string in the configuration file (log.conf).

Example:

```
########## # A new file is created every midnight or on reaching 20MB file size, # and the file name is /datadisk5/netscaler/log/NS<hostname>/Nsmmddyy.log and create digital #signature field for each record. BEGIN CACHE_F logFormat custom "%a - "%{
 user-agent }
 i" [%d/%B/%Y %T -%g] "%x" %s %b%{
 referrer }
 i "%{
 user-agent }
 i" "%{
 cookie }
 i" %d " logInterval Daily logFileSizeLimit 20 logFilenameFormat /datadisk5/netscaler/log/%v/NS%`{
 `%m%d%y }
 t.log END CACHE_F
<!--NeedCopy-->
```

Creating a Custom log format manually

To customize the format in which log file data should appear, specify a character string as the argument of the LogFormat log property definition. For more information, see Arguments for Defining a Custom Log Format. The following is an example where character strings are used to create a log format:

```
LogFormat Custom ""%a - "%{
 user-agent }
 i" %[%d/%m/%Y]t %U %s %b %T"
<!--NeedCopy-->
```

- The string can contain the "c" type control characters \n and \t to represent new lines and tabs.
- Use the Esc key with literal quotes and backslashes.

The characteristics of the request are logged by placing % directives in the format string, which are replaced in the log file by the values.

If the %v (Host name) or %x (URL suffix) format specifier is present in a log file name format string, the following characters in the file name are replaced by an underscore symbol in the log configuration file name:

```
”*./:<>?\
```

Characters whose ASCII values lie in the range of 0-31 are replaced by the following:

`%<ASCII value of character in hexadecimal>.`

For example, the character with ASCII value 22 is replaced by %16.

> **Caution**:
>
> If the %v format specifier is present in a log file name format string, a separate file is opened for each virtual host. To ensure continuous logging, the maximum number of files that a process can have open should be sufficiently large. See your operating system documentation for a procedure to change the number of files that can be opened.

Creating Apache log formats

You can derive from the custom logs most of the log formats that Apache currently supports. The custom log formats that match Apache log formats are:

NCSA/combined: LogFormat custom %h %l %u [%t] "%r" %s %B "%{referer}i" "%{user-agent}i"

NCSA/Common: LogFormat custom %h %l %u [%t] "%r" %s %B

Referer Log: LogFormat custom "%{referer}i" -> %U

Useragent: LogFormat custom %{user-agent}i

Similarly, you can derive the other server log formats from the custom formats.

Arguments for defining a custom log format

Refer to the Custom log format pdf table for more information about defining a custom log format.

For example, if you define the log format as %+{user-agent}i, and if the user agent value is NetScaler system Web Client, then the information is logged as NetScaler system+Web+Client. An alternative is to use double quotation marks. For example, "%{user-agent}i" logs it as "NetScaler system Web

Client.” Do not use the <Esc> key on strings from %.. .r, %. . .i and, %. . .o. This complies with the requirements of the Common Log Format. Note that clients can insert control characters into the log. Therefore, you should take care when working with raw log files.

Time format definition

Refer to the Time format definition table to know about the format part of the %{format}t string described in the Custom Log Format table. Values within brackets ([ ]) show the range of values that appear. For example, [1,31] in the %d description in the following table shows %d ranges from 1 to 31.

> **Note**:
>
> If you specify a conversion that does not correspond to any of the ones described in the preceding table, or to any of the modified conversion specifications listed in the next paragraph, the behavior is undefined and returns 0.

The difference between %U and %W (and also between modified conversions %OU and %OW) is the day considered to be the first day of the week. Week number 1 is the first week in January (starting with a Sunday for %U, or a Monday for %W). Week number 0 contains the days before the first Sunday or Monday in January for %U and %W.

Displaying Server Logs

You can configure a NetScaler Web Logging (NSWL) feature to display server logs on the console or redirect server logs to a directory on the NetScaler appliance.

There are two ways to display logs on the console (standard output):
Option 1: Display all logs on the console.
Option 2: Display only selected logs on the console where filters with logfilenameformat as STDOUT.

Call Home

September 21, 2020

Appliances might sometimes fail to perform well because of software or hardware issues. In such cases, Citrix needs to collect data and perform issue resolution before a potential impact can occur at the customer site. By enabling Call Home on your NetScaler appliance, you can automate the error notification process. Not only you avoid calling Citrix support, raising a service request, and uploading system data before the support team can troubleshoot the issue, but support can identify and address an issue before it occurs. Call Home periodically monitors the appliance and automatically uploads data to the Citrix technical support server. In addition, the incoming Call Home data provide insights about NetScaler usage. Multiple teams within Citrix can use this data to better design, support, and implement NetScaler.

By default, Call Home is enabled on all platforms and all flavors of NetScaler (MPX, VPX, SDX). By having this feature enabled, you allow Citrix to collect NetScaler deployment and telemetry data for better implementation, and support service.

> **Note**
>
> You can also see the Call Home FAQ page for information pertaining to Call Home.

Benefits

Call Home provides the following benefits.

- Monitor hardware and software error conditions. For more information, see Monitor critical error conditions section.
- Notify critical events that impact your network.
- Send performance data and system usage details to Citrix to:
  - Analyze and improve product quality.
  - Provide real-time troubleshooting information for proactive issue identification, and faster issue resolution.

Platform support

Call Home feature is supported on all NetScaler platforms and all appliance models (MPX, VPX, and SDX).

- NetScaler MPX: All MPX models.
- NetScaler VPX: All VPX models, including VPX appliances that obtain their license from external or central licensing pools.
- NetScaler SDX: Monitors the disk drive and assigned SSL chips for any errors or failures. The VPX instances, however, do not have access to the Power Supply Unit (PSU) and therefore their status is not monitored. In an SDX platform, you can configure Call Home either directly on an individual instance or through the SVM.

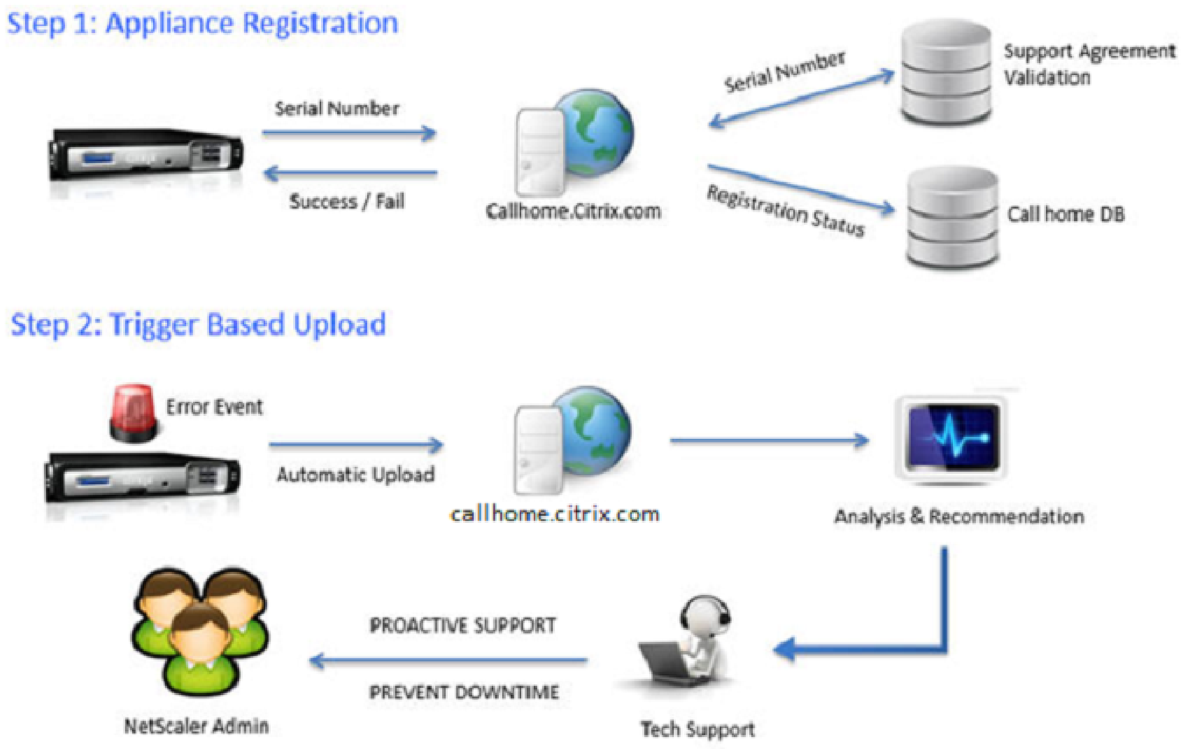
Prerequisites

To use Call Home, the NetScaler appliance must have the following:

- **Internet connection.** Call Home requires an internet connection for NetScaler to connect to the Citrix support server for uploading a data archive.

How Call Home Works

The following figure shows the basic workflow of Call Home in a NetScaler appliance deployed at a customer site.


Step 1: Appliance Registration
Serial Number
Success / Fail
Callhome.Citrix.com
Serial Number
Support Agreement Validation
Registration Status
Call home DB
Step 2: Trigger Based Upload
Error Event
Automatic Upload
callhome.citrix.com
Analysis & Recommendation
PROACTIVE SUPPORT
PREVENT DOWNTIME
NetScaler Admin
Tech Support

The following are the workflow of a Call Home:

1. Set up Internet Connectivity. For Call Home to upload system data, your appliance must have an internet connectivity. If it does not, you can configure a proxy server configuration to provide internet connectivity. For more information, see Configuring Call Home section.

2. Enable Call Home. When upgrading your appliance to the latest software through either NetScaler command interface or GUI, Call Home is enabled by default and the system delays the registration process by 24 hours. During this period, you can choose to manually disable the feature, but Citrix recommends you to have it enabled.

> **Note**
>
> If you are upgrading your appliance from an older version that explicitly has Call Home disabled, the system still enables the feature by default and displays a notification message on your first login.

In addition, if you are doing any configuration changes for an internet connectivity, you must disable and enable Call Home. It enables Call Home to register with the Citrix Insight Services (CIS) server without any failure errors.

3. Register the NetScaler appliance on the Citrix Support Server. When Call Home registers the appliance with the Citrix Support server, the server checks the database for the validity of the appliance serial number. If the serial number is valid, the server registers the appliance for Call Home service and sends a successful registration response. Otherwise, the server sends back a registration failure message. The basic system information is sent as a separate message. The data includes memory and CPU usage details along with the throughput numbers. The data is sent periodically as part of the heartbeat message every 30 days, by default. However, a value of less than 5 days is not recommended, because frequent uploads are not useful.

4. Monitor critical error conditions. Once registered, Call Home starts monitoring the appliance. The following table lists the conditions Call Home can monitor on the appliance.

Critical Error Condition	Description	Call Home Monitoring Interval	Corresponding SNMP Alarm Name
Compact flash drive errors	The compact flash drive on the appliance encountered read or write failures.	24 hours	COMPACT-FLASH-ERRORS
Hard disk drive errors	The hard drives on the appliance encountered read or write failures.	24 hours	HARD-DISK-DRIVE-ERRORS
Power supply unit failure	One of the power supply units on the NetScaler appliance has failed.	7 seconds	POWER-SUPPLY-FAILURE
SSL card failure	One of the SSL cards on the NetScaler appliance has failed.	7 seconds	SSL-CARD-FAILED
Warm restart	The appliance has warm restarted due to a failure of a system process.	After every restart of the NetScaler appliance.	WARM-RESTART-EVENT

Critical Error Condition	Description	Call Home Monitoring Interval	Corresponding SNMP Alarm Name
Memory anamoly error	Memory utilization progressively increases above its normal limit and exceeds the threshold.	1 day	No SNMP alarm
Rate limit packet drop	The throughput limits or packets-per-second (pps) limits is reached.	7 seconds	PF-RL-PPS-PKTS-DROPPED, PF-RL-RATE-PKTS-DROPPED

5. **Upload Call Home data.** If any one of the previous critical conditions is identified on the appliance, the Call Home feature automatically notifies the Citrix support. The support archives are uploaded to the Citrix support server. Also, you can configure the CALLHOME-UPLOAD-EVENT SNMP alarm to generate an SNMP alert whenever Call Home upload happens. The SNMP alert notifies the local administrator about the critical event.

> **Note**
>
> Call Home creates the Call Home tar file and uploads it to the Citrix tech support server for only the first occurrence of a particular error condition since the last reboot. If you want the appliance to send alerts each time a particular error condition occurs, configure the corresponding SNMP alarm for the error condition.

6. **Create Service Request**. Call Home automatically creates a service request for the critical hardware related events. The events are classified as; power supply failure, SSL card failure, hard disk drive errors, and compact flash errors. For other errors, after you review the System Logs, you can contact the Citrix support team to raise a service request for investigation.

Configuring Call Home

To configure Call Home, verify the internet connectivity on the appliance and ensure a DNS nameserver is configured. If there is no internet connection, configure a proxy server or service. Then, enable Call Home on the appliance and verify the registration status of the appliance with a Citrix support server. Once registered, Call Home can monitor and upload data. In addition, you can configure SNMP alarms to notify the administrator at the customer site.

To configure Call Home, you can use either the NetScaler command interface or the GUI to do the following tasks:

- Enable Call Home.
- Configure Call Home for optional proxy server parameters.
- Verify Call Home Registration Status.
- View errors and timestamp details.
- Configure SNMP Alarms.

To configure Call Home by using the NetScaler command interface

The NetScaler command interface enables you to do the following:

`Enabling Call Home`

At the command prompt, type:

`enable ns feature callhome`

Configuring Call Home for optional proxy server parameters

Call Home enables you to configure the optional proxy server for internet connectivity. You can either configure a proxy server with IP address and port or configure a proxy authentication service with one-way or two-way authentication.

`To configure optional proxy server with IP address and port`

At the command prompt, type:

`set callhome -proxyMode ( YES | NO )[-IPAddress <ip_addr|ipv6_addr|*>] [-port <port |*>]`

```
set callhome -proxyMode YES -IPAddress 10.102.167.33 -port 80
<!--NeedCopy-->
```

> **Note**
>
> Call Home uses the proxy server only when you set the proxy-mode parameter to YES. If you set it to NO, the proxy functionality does not work, even if the IP address and port are configured. The port number must be for an HTTP service, not for an HTTPS service.

To configure an optional proxy authentication service:

This mode provides two types of security authentication: one-way and two-way. To set up either type, you must configure an SSL service. For more information, see Configuring an SSL Service topic.

In one-way authentication, only the NetScaler appliance authenticates the proxy server. In two-way authentication, the NetScaler appliance authenticates the proxy server and the proxy server, in turn, authenticates the appliance.

To configure proxy authentication service

At the command prompt, type:

```
set callhome -proxyMode ( YES | NO )[-proxyAuthService <string>]
```

```
set callhome  - proxyMode YES  - proxyAuthService callhome_proxy
<!--NeedCopy-->
```

To configure one-way proxy server authentication

Do the following tasks to configure one-way proxy server authentication.

1. Create an SSL service.
2. Bind a CA certificate to the service.
3. Bind an HTTPS Monitor to the service.
4. Configure Call Home to use the SSL service.

To configure two-way proxy server authentication

Do the following tasks to configure two-way proxy server authentication.

1. Create an SSL service
2. Bind a CA certificate to the service.
3. Bind a Client Certificate.
4. Bind an HTTPS Monitor to the service.
5. Configure Call Home to use the SSL service.

Verifying Call Home Registration status

At the command prompt, type:

```
show callhome

    show callhome

     Registration with Citrix upload server SUCCESSFUL

     Mode: Default

     Contact email address: exampleadmin@example.com

     Heartbeat Custom Interval (days): 30

     Proxy Mode: Yes

        Proxy IP Address:10.102.29.200

        Proxy Authentication Service:

        Proxy Port: 80

    Trigger event                  State   First
       occurrence                Latest occurrence

     -------------                  -----   ----------------
                       -----------------

     1) Warm boot                   Enabled N/A
                        ..

     2) Compact flash errors        Enabled ..
                                     ..

     3) Hard disk drive errors      Enabled ..
                                     ..

     4) SSL card failure            N/A     N/
       A                             N/A

     5) Power supply unit failure   N/A     N/
       A                             N/A

     6) Rate limit packet drops     Enabled ..
                                     ..

     7) Memory anomaly              Enabled ..
                                     ..

     Done
<!--NeedCopy-->
```

> **Note**
>
> If the Call Home fails to register with CIS, the appliance displays an error message.

Enabling SNMP Alarms

The NetScaler appliance provides a set of error condition entities called *SNMP alarms*. When an error condition in an SNMP alarm is met, the appliance generates SNMP trap messages that are sent to the configured trap listeners. For example, when the SSL-CARD-FAILED alarm is enabled, a trap message is generated and sent to the trap listener whenever there is an SSL card failure on the appliance. For more information, see SNMP.

At the command prompt, type:

```
enable snmp alarm <trapName>
```

```
show snmp alarm <trapName>
```

To configure Call Home by using GUI

The GUI enables you to do the following:

To verify if the Call Home feature is enabled by default in the GUI

1. Navigate to **Configuration** > **System** > **Settings**.
2. In the **details** pane, click **Configure Advanced features** link.
3. In the **Configure Advanced Features** page, **Call Home** option must display as enabled.

To enable Call Home by using GUI

1. Navigate to **Configuration** > **System** > **Settings**.
2. In the **details** pane, click **Configure Advanced features** link and select **Callhome** option.

To configure Call Home for optional proxy mode authentication by using GUI

1. You can use any of the two ways to access the Call Home page:
   a) Navigate to **System** > **System Information**.
   b) Navigate to **System** > **Diagnostics**.
      i. In the details pane, under **Technical Support Tools**, select **Call Home**.
2. On the **Configure Call Home** page, set the following parameters.
   a) **Mode**. Call Home mode of operation. Possible types: Default, Citrix Service Provider (CSP) deployment.
   b) **Email address**. Email address of the contact administrator at the customer site.
   c) **CallHome Heartbeats Interval (days).** Monitoring interval (in days) between Call Home heartbeats. Minimum value=1 and Maximum value=30.
   d) **Enable Call Home**. Enable or disable the Call Home feature to view the status of appliance registration on the Citrix support server.
   e) **Proxy Mode**. If you do not have an internet connectivity, enable proxy mode and set the optional proxy parameters.
   f) **Proxy Server**. If you set the proxy mode by using a proxy server, specify the server IP address.
      i. **Proxy Service**. If you set the proxy mode by using a proxy service, specify the service name.
      ii. **IP Address**. IP address of the proxy server.
      iii. **Port**. Port number of the proxy server.
      iv. **Proxy Authentication SSL Service**. The name of the proxy service that provides proxy-mode authentication.

3. Click **OK** and **Done**.

To configure SSL service for proxy server authentication by using GUI

For information about configuring an SSL service by using the GUI, see Configuring an SSL Service topic.

To verify Call Home registration status by using the GUI

1. You can use any of the two ways to access the **Call Home** page:
    a) Navigate to **System** > **System Information**.
    b) Navigate to **System** > **Diagnostics**.
        i. In the details pane, under **Technical Support Tools**, select **Call Home**.
2. In the **Configure Call Home** page, the **Registration with Citrix upload server** field shows the registration status.

To configure an SNMP alarm

1. Navigate to **System** > **SNMP** > **Alarms**.
2. In the details pane, select an alarm and configure its parameters.
3. Click **OK** and **Close**.

Citrix Service Provider (CSP) deployment support

In a Citrix Service Provider (CSP) environment where NetScaler services are deployed on VPX instances, Call Home can monitor and track the license-specific information and securely send the information to Citrix Insight Services (CIS). CIS in turn sends the information to the License Usage Insights (LUI) portal for accounting purposes and for CSP customers to review their license usage. Currently, CSP environments support NetScaler services on VPX instances only, not on MPX or SDX appliances. The VPX instances can be deployed in either standalone or high availability mode.

Reporting tool

September 21, 2020

Use the Citrix® NetScaler® Reporting tool to view NetScaler performance statistics data as reports. Statistics data are collected by the nscollect utility and are stored in a database. When you want to view certain performance data over a period of time, the Reporting tool pulls out specified data from the database and displays them in charts.

Reports are a collection of charts. The Reporting tool provides built-in reports as well as the option to create custom reports. In a report, you can modify the charts and add new charts. You can also modify the operation of the data collection utility, nscollect, and stop or start its operation.

Using the reporting tool

The Reporting tool is a Web-based interface accessed from the Citrix® NetScaler® appliance. Use the Reporting tool to display the performance statistics data as reports containing graphs. In addition to using the built-in reports, you can create custom reports, which you can modify at any time. Reports can have between one and four charts. You can create up to 256 custom r

To invoke the reporting tool

1. Use the Web browser of your choice to connect to the IP address of the NetScaler (for example, `http://10.102.29.170/`). The Web Logon screen appears.
2. In the User Name text box, type the user name assigned to the NetScaler.
3. In the Password text box, type the password.
4. In the Start in drop-down box, select Reporting.Click Login.

The following screen shots show the report toolbar and the chart toolbar, which are frequently referenced in this documentation.

Figure 1. *Report Toolbar*

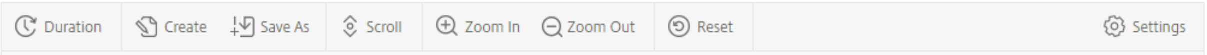
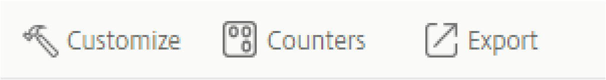


Figure 2. *Chart Toolbar*



Working with reports

You can plot and monitor statistics for the various functional groups configured on the NetScaler over a specified time interval. Reports enable you to troubleshoot or analyze the behavior of your appliance. There are two types of reports: built-in reports and custom reports. Report content for built-in or custom reports can be viewed in a graphical format or a tabular format. The graphical view consists of line, area, and bar charts that can display up to 32 sets of data (counters). The tabular view displays the data in columns and rows. This view is useful for debugging error counters.

The default report that is displayed in the Reporting tool is CPU vs. Memory Usage and HTTP Requests Rate. You can change the default report view by displaying the report you want as your default view, and then clicking Default Report.

Reports can be generated for the last hour, last day, last week, last month, last year, or you can customize the duration.

You can do the following with reports:

- Toggle between a tabular view of data and a graphical view of data.

- Change the graphical display type, such as bar chart or line chart.
- Customize charts in a report.
- Export the chart as an Excel comma-separated value (CSV) file.
- View the charts in detail by zooming in, zooming out, or using a drag-and-drop operation (scrolling).
- Set a report as the default report for viewing whenever you log on.
- Add or remove counters.
- Print reports.
- Refresh reports to view the latest performance data.

Using built-in reports

The Reporting tool provides built-in reports for frequently viewed data. Built-in reports are available for the following functional groups: System, Network, SSL, Compression, Integrated Cache, NetScaler Gateway, and NetScaler Application Firewall. By default, the built-in reports are displayed for the last day. However, you can view the reports for the last hour, last week, last month, or last year.

> **Note**:
>
> You cannot save changes to built-in reports, but you can save a modified built-in report as a custom report.

To display a built-in report

1. In the left pane of the Reporting tool, under Built-in Reports, expand a group (for example, SSL).
2. Click a report (for example, SSL > All Backend Ciphers).

Creating and deleting reports

You can create your own custom reports and save them with user-defined names for reuse. You can plot different counters for different groups based on your requirements. You can create up to 256 custom reports.

You can either create a new report or save a built-in report as a custom report. By default, a newly created custom report contains one chart named System Overview, which displays the CPU Usage counter plotted for the last day. You can customize the interval and set the data source and time zone from the report toolbar. Within a report, you can use the chart toolbars to add, modify, or delete charts, as described in “Working with Charts.”

By default, newly created custom reports contain one chart named System Overview that displays a CPU Usage counter plotted for the last day.

To create a custom report

1. In the **Reporting** tool, on the report toolbar, click **Create**, or if you want to create a new custom report based on an existing report, open the existing report, and then click **Save As**.
2. In **Report Name** box, type a name for the custom report.
3. Do one of the following:
    - To add the report to an existing folder, in Create in or Save in, click the down arrow to choose an existing folder, and then click **OK**.
    - To create a new folder to store the report, click the Click to add folder icon, in Folder Name, type the name of the folder, and in Create in, specify where you want the new folder to reside in the hierarchy, and then click **OK**.

> **Note**:
> You can create up to 128 folders.

To delete a custom report

1. In the left pane of the Reporting tool, next to Custom Reports, click the Click to manage custom reports icon.

2. Select the check box that corresponds with the report you want to delete, and then click Delete.

    > **Note**:
    >
    > When you delete a folder, all the contents of that folder are deleted.

Modifying the time interval

By default, built-in reports display data for the last day. However, if you want to change the time interval for a built-in report, you can save the report as a custom report. The new interval applies to all charts in the report. The following table describes the time-interval options.

To modify the time interval

1. In the left pane of the Reporting tool, click a report.
2. On the report toolbar, click **Duration**, and then click a time interval.

Setting the data Source and time zone

You can retrieve data from different data sources to display them in the reports. You can also define the time zone for the reports and apply the currently displayed report's time selection to all the reports, including the built-in reports.

To set the data source and time zone

1. In the **Reporting tool**, on the report toolbar, click **Settings**.
2. In the **Settings** dialog box, in Data Source, select the data source from which you want to retrieve the counter information.
3. Do one or both of the following:
   - If you want the tool to remember the time period for which a chart is plotted, select the **Remember time selection for charts** check box.
   - If you want the reports to use the time settings of your NetScaler appliance, select the **Use Appliance's time zone** check box.

Exporting and importing custom reports

You can share reports with other NetScaler administrators by exporting reports. You can also import reports.

To export or import custom reports

1. In the left pane of the Reporting tool, next to Custom Reports, click the **Click to manage custom reports** icon.
2. Select the check box that corresponds with the report you want to export or import, and then click **Export** or **Import**.

   > **Note**:
   >
   > When you export the file, it is exported in a .gz file format.

Working with charts

Use charts to plot and monitor counters or groups of counters. You can include up to four charts in one report. In each chart, you can plot up to 32 counters. The charts can use different graphical formats (for example, area and bar). You can move the charts up or down within the report, customize the colors and visual display for each counter in a chart, and delete a chart when you do not want to monitor it.

In all report charts, the horizontal axis represents time and the vertical axis represents the value of the counter.

Adding a chart

When you add a chart to a report, the System Overview chart appears with the CPU Usage counter plotted for the last one day. To plot a different group of statistics or select a different counter, see "Modifying a Chart."

> **Note**:
>
> If you add charts to a built-in report, and you want to retain the report, you must save the report as a custom report.

Use the following procedure to add a chart to a report.

To add a chart to a report

1. In the left pane of the Reporting tool, click a report.
2. Under the chart where you want to add the new chart, click the Add icon.

Modifying a chart

You can modify a chart by changing the functional group for which the statistics are displayed and by selecting different counters.

To modify a chart

1. In the left pane of the Reporting tool, click a report.
2. Under the chart that you want to modify, click Counters.
3. In the dialog box that appears, in the Title box, type a name for the chart.
4. Next to Plot chart for, do one of the following:
    - To plot counters for global counters, such as Integrated Cache and Compression, click System global statistics.
    - To plot entity counters for entity types, such as Load Balancing and GSLB, click System entities statistics.
5. In Select group, click the desired entity.
6. Under Counters, in Available, click the counter name(s) that you want to plot, and then click the **>** button.
7. If you selected System entities statistics in step 4, on the Entities tab, under Available, click the entity instance name(s) you want to plot, and then click the > button.
8. Click OK.

Viewing a chart

You can specify the graphical formats of the plotted counters in a chart. Charts can be viewed as line charts, spline charts, step-line charts, scatter charts, area charts, bar charts, stacked area charts, and stacked bar charts. You can also zoom in, zoom out, or scroll inside the plot area of a chart. You can zoom in or out for all data sources for 1 hour, 1 day, 1 week, 1 month, 1 year, and 3 years.

Other options for customizing the view of a chart include customizing the axes of the charts, changing the background and edge color of the plot area, customizing the color and size of the grids, and customizing the display of each data set (counter) in a chart.

Data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.

Whenever you modify a built-in report, you need to save the report as a custom report to retain your changes.

To change the graph type of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart you want to view, on the chart toolbar, click **Customize**.
3. On the **Chart** tab, under **Category**, click **Plot type**, and then click the graph type you want to display for the chart. If you want to display the graph is 3D, select the Use 3D check box.

To refocus a chart with detailed data

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Zoom In**, and do one or both of the following:
   - To refocus the chart to display data for a specific time window, drag and drop the cursor from the start time to the end time. For example, you can view data for a one-hour period on a certain day.
   - To refocus the chart to display data for a data point, simply click once on chart where you want to zoom in and get more detailed information.
3. Once you have the desired range of time for which you want to view detailed data, on the report toolbar, click Tabular View. Tabular view displays the data in numeric form in rows and columns.

To view numeric data for a graph

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Tabular View**. To return to the graphical view, click **Graphical View**.

   > **Note**: You can also view the numeric data in the graphical view by hovering your cursor over the notches in the gridlines.

To scroll through time in a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, on the report toolbar, click **Scroll**, and then click inside the chart and drag the cursor in the direction for which you want to see data for a new time period. For example, if you want to view data in the past, click and drag to the left.

To change the background color and text color of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under **Category**, click one or more of the following:
    - To change the background color, click **Background Color**, and then select the options for color, transparency, and effects.
    - To change the text color, click Text **Color**, and then select the options for color, transparency, and effects.

To customize the axes of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the axes, click **Customize**.
3. On the **Chart** tab, under Category, click one or more of the following:
    - To change the scale of the left y-axis, click **Left Y-Axis**, and then select the scale you want.
    - To change the scale of the right y-axis, click **Right Y-Axis**, in Data set to plot, select the date set, and then select the scale you want.

      > Note:
      > The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if CPU usage and Memory usage are displayed in first and second order at the bottom of the chart, CPU usage is equal to Data Set 1 and Memory usage is equal to Data Set 2.

    - To plot each data set in its own hidden y-axis, click Multiple Axes, and then click **Enable**.

To change the background color, edge color, and gridlines for a plot area of a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the plot area, click **Customize**.
3. On the **Plot Area** tab, under Category, click one or more of the following:
    - To change the background color and edge color of the chart, click **Background Color and Edge Color**, and then select the options for color, transparency, and effects.

  - To change the horizontal or vertical grids of the chart, click **Horizontal Grids** or **Vertical Grids**, and then select the options for displaying the grids, grid width, grid color, transparency, and effects.

To change the color and graph type of a data set

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart for which you want to customize the display of the data set (counters), click **Customize**.
3. On the **Data Set** tab, in Select Data Set, select the data set (counter) for which you want to customize the graphical display.
   Note: The data set numbers, such as Data Set 1, correspond to the order in which the counters in your graph are displayed at the bottom of the chart. For example, if
   CPU usage and
   Memory usage are displayed in first and second order at the bottom of the chart,
   CPU usage is equal to
   Data Set 1 and
   Memory usage is equal to
   Data Set 2.
4. Under Category, do one of more of the following:
   - To change the background color, click **Color**, and then select the options for color, transparency, and effects.
   - To change the graph type, click **Plot type**, and then select the graph type you want to display for the data set. If you want to display the graph as 3D, select the Use 3D check box.

Exporting chart data to excel

For further data analysis, you can export charts to Excel in a comma-separated value (CSV) format.

To export chart data to excel

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart with the data you want to export to Excel, click **Export**.

Deleting a chart

If you do not want to use a chart, you can remove it from the report. You can permanently remove charts from custom reports only. If you delete a chart from a built-in report and want to retain the changes, you need to save the report as a custom report.

To delete a chart

1. In the left pane of the Reporting tool, select a report.
2. In the right pane, under the chart that you want to delete, click the **Delete** icon.

Examples

To display the trend report for CPU usage and memory usage for the last week

1. In the left pane of the Reporting tool, under Built-in Reports, expand System.
2. Click the report CPU vs. Memory Usage and HTTP Requests Rate.
3. In the right pane, on the report toolbar, click **Duration**, and then click **Last Week**.

To compare the bytes received rate and the bytes transmitted rate between two interfaces for the last week

1. In the right pane, on the report toolbar, click Create.
2. In the **Report Name** box, type a name for the custom report (for example, Custom_Interfaces), and then click **OK**. The report is created with the default System Overview chart, which displays the CPU Usage counter plotted for the last hour.
3. Under System Overview, on the chart toolbar, click **Counters**.
4. In the counter selection pane, in Title, type a name for the chart (for example, Interfaces bytes data).
5. In Plot chart for, click System entities statistics, and then in Select Group, select **Interface**.
6. On the **Entities** tab, click the interface name(s) you want to plot (for example, 1/1 and 1/2), and then click the **>** button.
7. On the **Counters** tab, click Bytes received (Rate) and Bytes transmitted (Rate) and then click the > button.
8. Click **OK**.
9. On the report toolbar, click **Duration**, and then click **Last Week**.

Stopping and starting the data collection utility

The data collection utility, nscollect, runs automatically when you start the NetScaler. This utility retrieves the application performance data and stores it in the form of data sources on the ADC. You can create up to 32 data sources. The default data source is `/var/log/db/default`.

The data collection utility creates databases for global counters and entity-specific counters, and uses this data to generate reports. Global-counter databases are created at `/var/log/db/<DataSourceName>`. The entity-specific databases are created based on the entities configured on the NetScaler, and a separate folder is created for each entity type in `/var/log/db/<DataSourceName/EntityNameDB>`.

Nscollect retrieves data once every 5 minutes. It retains data in 5-minute granularity for one day, hourly for the last 30 days, and daily for three years.

You might have to stop and restart the data collection utility if data is not updated accurately or the reports display corrupted data.

To stop nscollect

At the command prompt, type:

```
/netscaler/nscollect stop
```

To start nscollect on the current SSH session to the NetScaler:

At the command prompt, type:

```
/netscaler/nscollect start
```

To start nscollect on the local system:

At the command prompt, type:

```
/netscaler/nscollect start &
```

CloudBridge Connector

June 29, 2021

Note: The current NetScaler 1000V release does not support this feature.

The CloudBridge Connector feature of the NetScaler appliance connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of your enterprise network. Cloud-hosted applications appear as though they are running on one contiguous enterprise network. With Citrix CloudBridge Connector, you can augment your datacenters with the capacity and efficiency available from cloud providers.

The CloudBridge Connector enables you to move your applications to the cloud to reduce costs and increase reliability.

In addition to using CloudBridge Connector between a datacenter and a cloud, you can use it to connect two datacenters for a high-capacity secure and accelerated link.

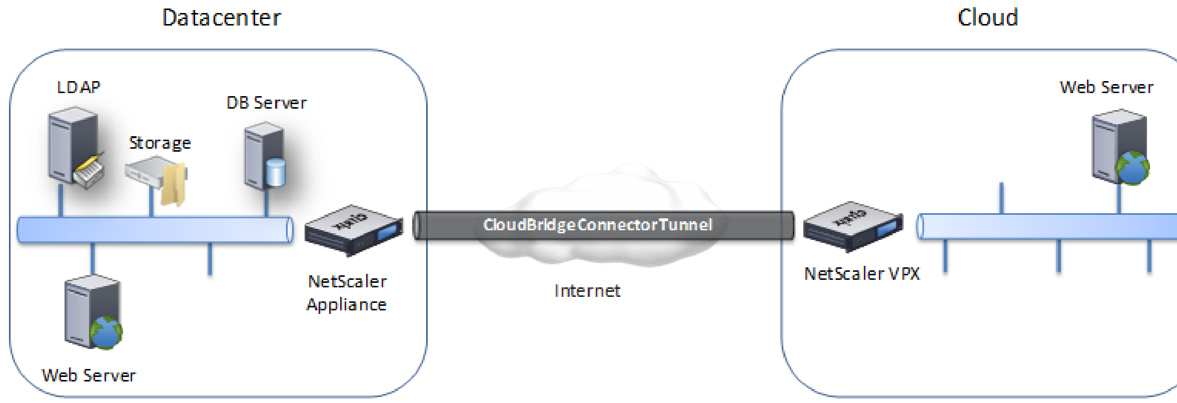
Understanding CloudBridge Connector

To implement the Citrix CloudBridge Connector solution, you connect a datacenter to another datacenter or an external cloud by setting up a tunnel called the CloudBridge Connector tunnel.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between two NetScaler appliances, one in each datacenter.

To connect a datacenter to an external cloud (for example, Amazon AWS cloud), you set up a CloudBridge Connector tunnel between a NetScaler appliance in the datacenter and a virtual appliance (VPX) that resides in the Cloud. The remote end point can be a CloudBridge Connector or a NetScaler VPX with platinum license.

The following illustration shows a CloudBridge Connector tunnel set up between a datacenter and an external cloud.



The appliances between which a CloudBridge Connector tunnel is set up are called the *end points* or *peers* of the CloudBridge Connector tunnel.

A CloudBridge Connector tunnel uses the following protocols:

- Generic Routing Encapsulation (GRE) protocol
- Open-standard IPSec Protocol suite, in transport mode

The GRE protocol provides a mechanism for encapsulating packets, from a wide variety of network protocols, to be forwarded over another protocol. GRE is used to:

- Connect networks running non-IP and non-routable protocols.
- Bridge across a wide area network (WAN).
- Create a transport tunnel for any type of traffic that needs to be sent unchanged across a different network.

The GRE protocol encapsulates packets by adding a GRE header and a GRE IP header to the packets.

The Internet Protocol security (IPSec) protocol suite secures communication between peers in the CloudBridge Connector tunnel.

In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication

- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the transport mode in which the GRE encapsulated packet is encrypted. The encryption is done by the Encapsulating Security Payload (ESP) protocol. The ESP protocol ensures the integrity of the packet by using a HMAC hash function, and ensures confidentiality by using an encryption algorithm. After the packet is encrypted and the HMAC is calculated, an ESP header is generated. The ESP header is inserted after the GRE IP header and, an ESP trailer is inserted at the end of the encrypted payload.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version (IKE) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

- The two peers mutually authenticate with each other, using one of the following authentication methods:
  - **Pre-shared key authentication**. A text string called a pre-shared key is manually configured on each peer. The pre-shared keys of the peers are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
  - **Digital certificates authentication**. The initiator (sender) peer signs message interchange data by using its private key, and the other receiver peer uses the sender's public key to verify the signature. Typically, the public key is exchanged in messages containing an X.509v3 certificate. This certificate provides a level of assurance that a peer's identity as represented in the certificate is associated with a particular public key.
- The peers then negotiate to reach agreement on:
  - An encryption algorithm.
  - Cryptographic keys for encrypting data in one peer and decrypting the data in the other.

This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when two peers, CB1 and CB2, are communicating through a Connector tunnel, CB1 has two Security Associations. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets.

SAs expire after a specified length of time, which is called the *lifetime*. The two peers use the Internet Key Exchange (IKE) protocol (part of the IPSec protocol suite) to negotiate new cryptographic keys and establish new SAs. The purpose of the limited lifetime is to prevent attackers from cracking a key.

The following table lists some IPSec propeties supported by a NetScaler appliance:

IPSec Properties	Types Supported
IKE Versions	V1, V2
IKE DH group	A Citrix NetScaler appliance supports only DH group 2 (1024 bits MODP algorithm) for both IKEv1 and IKEv2.
IKE Authentication Methods	Pre-shared key authentication, Digital certificates authentication
Encryption Algorithm	AES (128 bits), AES 256 (256 bits), 3DES
Hash Algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5

Monitoring CloudBridge Connector tunnels

September 21, 2020

You can display the statistics for monitoring the performance of a CloudBridge Connector tunnel. To display CloudBridge Connector tunnel statistics on a NetScaler appliance, use the GUI or the NetScaler command line.

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels on a NetScaler appliance.

Statistical counter	**Specifies**
Bytes Received	Total number of bytes received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Sent	Total number of bytes sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Packets Received	Total number of packets received by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.

Packets Sent	Total number of packets sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels since the appliance was last started.
Bytes Received Rate	Number of bytes per second received by the NetScaler appliance through all the configured CloudBridge Connector tunnels.
Bytes Sent Rate	Number of bytes per second sent by the NetScaler appliance through all the configured CloudBridge Connector tunnels
Packets Received Rate	Number of bytes per second received by the NetScaler appliance through all the configured CloudBridge Connector tunnels
Packets Sent Rate	Number of bytes per second received by the NetScaler appliance through all the configured CloudBridge Connector tunnels

All these counters are reset to 0 when the NetScaler appliance is restarted. They do not increment during the following phases:

- Internet Key Exchange (IKE) authentication (pre-shared key) phase on any configured CloudBridge Connector tunnel.
- IKE Security Association (SA) establishment phase on any configured CloudBridge Connector tunnel.

To display CloudBridge Connector tunnel statistics by using the NetScaler command line

At the command prompt, type:

- **stat ipsec counters**

To display CloudBridge Connector tunnel statistics by using the GUI

1. Access the GUI by using a web browser to connect to the IP address of the NetScaler appliance.
2. On the **Configuation** tab, navigate to **System** > **CloudBridge Connector**.
3. On the CloudBridge Connector page, click **Create/Monitor CloudBridge Connector**. The **IPSec Bytes** and **IPSec Packets** charts display the bytes received rate, bytes sent rate, packets received rate, and packets sent rate of all the CloudBridge Connector tunnels configured on the NetScaler appliance.

```
    > stat ipsec counters
    Secure tunnel(s) summary
                        Rate (/s)        Total
    Bytes Received    0    2811248
    Bytes Sent            0    157460630
    Packets Received    0    56787
    Packets Sent        0    200910
    Done
    >
<!--NeedCopy-->
```

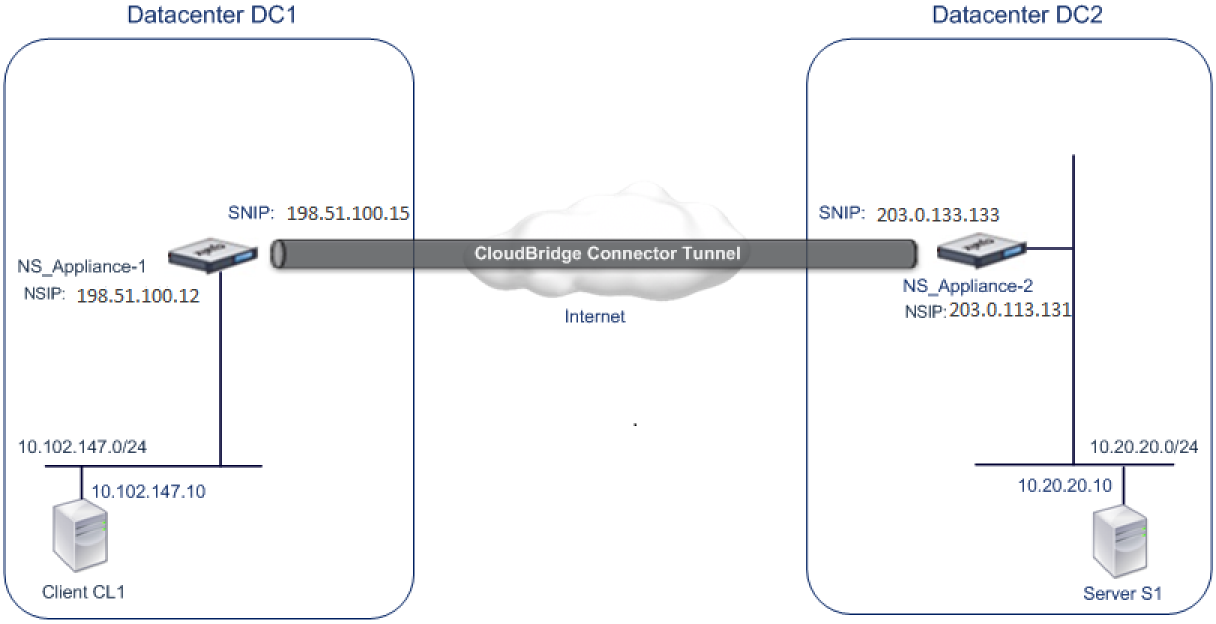
Configuring a CloudBridge Connector tunnel between two datacenters

September 21, 2020

You can configure a CloudBridge Connector tunnel between two different datacenters to extend your network without reconfiguring it, and leverage the capabilities of the two datacenters. A CloudBridge Connector tunnel between the two geographically separated datacenters enables you to implement redundancy and safeguard your setup from failure. The CloudBridge Connector tunnel helps achieve optimal utilization of infrastructure and resources across datacenters. The applications available across the two datacenters appear as local to the user.

To connect a datacenter to another datacenter, you set up a CloudBridge Connector tunnel between a NetScaler appliance in one datacenter and a NetScaler appliance in the other datacenter.

As an illustration of CloudBridge Connector tunnel between datacenters, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance NS_Appliance-1 in datacenter DC1 and NetScaler appliance NS_Appliance-2 in datacenter DC2.


Datacenter DC1
Datacenter DC2
SNIP: 198.51.100.15
NS_Appliance-1
NSIP: 198.51.100.12
CloudBridge Connector Tunnel
Internet
SNIP: 203.0.133.133
NS_Appliance-2
NSIP: 203.0.113.131
10.102.147.0/24
10.102.147.10
Client CL1
10.20.20.0/24
10.20.20.10
Server S1

Both NS_Appliance-1 and NS_Appliance-2 function in L2 and L3 mode. They enable communication between private networks in datacenters DC1 and DC2. In L3 mode, NS_Appliance-1 and NS_Appliance-2 enable communication between client CL1 in datacenter DC1 and server S1 in the datacenter DC2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

Because client CL1 and server S1 are on different private networks, L3 mode is enabled on NS_Appliance-1 and NS_Appliance-2, and routes are updated as follows:

- CL1 has a route to NS_Appliance-1 for reaching S1.
- NS_Appliance-1 has a route to NS_Appliance-2 for reaching S1.
- S1 has a route to NS_Appliance-2 for reaching CL1.
- NS_Appliance-2 has a route to NS_Appliance-1 for reaching CL1.

The following table lists the settings on NetScaler appliance NS_Appliance-1 in datacenter DC1.

The following table lists the settings on NetScaler appliance NS_Appliance-2 in datacenter DC2.

Entity	Name	Details
The NSIP address		198.51.100.12
SNIP address		198.51.100.15

Entity	Name	Details
CloudBridge Connector tunnel	Cloud_Connector_DC1-DC2	1. Local endpoint IP address of the CloudBridge Connector tunnel: 198.51.100.15, 2. Remote endpoint IP address of the CloudBridge Connector tunnel: 203.0.113.133. GRE Tunnel Details Name = Cloud_Connector_DC1-DC2, IPSec Profile Details Name = Cloud_Connector_DC1-DC2, Encryption algorithm = AES, Hash algorithm = HMAC SHA1

Points to consider for configuring CloudBridge Connector tunnel

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Deploy and set up a NetScaler appliance in each of the two datacenters.
2. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

Configuration procedure

To set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in one datacenter and another NetScaler appliance that resides in the other datacenter, use the GUI or the command line interface of one of the NetScaler appliances.

When you use the GUI, the CloudBridge Connector tunnel configuration created on the first NetScaler appliance is automatically pushed to the other endpoint (the other NetScaler appliance) of the CloudBridge Connector tunnel. Therefore, you do not have to access the GUI of the other NetScaler appliance to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on each of the NetScaler appliances consists of the following entities:

- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in the CloudBridge Connector tunnel.

- **GRE tunnel**—An IP tunnel specifies the local IP address (a public SNIP address configured on the local NetScaler appliance), remote IP address (a public SNIP address configured on the remote NetScaler appliance), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.
- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the first datacenter and is destined to a server on the subnet in the second datacenter. If this packet matches the source and destination IP address range of the PBR entity on the NetScaler appliance in the first datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

- `add ipsec profile <name> [-ikeVersion (V1 | V2)] [-encAlgo (AES | 3 DES)...] [-hashAlgo <hashAlgo\> ...] [-lifetime <positive_integer>] (-psk | (-publickey<string> -privatekey <string>-peerPublicKey <string >))[-livenessCheckInterval <positive_intege>][-replayWindowSize \< positive_integer>] [-ikeRetryInterval <positive_integer>] [-retransmissiontime <positive_integer>]`
- `show ipsec profile <name>`

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

At the command prompt, type:

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol < protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = < remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

Example

```
add ipsec profile Cloud_Connector_DC1-DC2  -encAlgo AES -
   hashAlgo HMAC_SHA1
```

```
    Done
    > add ipTunnel Cloud_Connector_DC1-DC2 203.0.113.133 255
       .255.255.255 198.51.100.15 -protocol GRE -
       ipsecProfileName Cloud_Connector_DC1-DC2

    Done
    > add ns pbr PBR-DC1-DC2 ALLOW -srcIP 198.51.100.15 -destIP
       203.0.113.133 ipTunnel Cloud_Connector_DC1-DC2

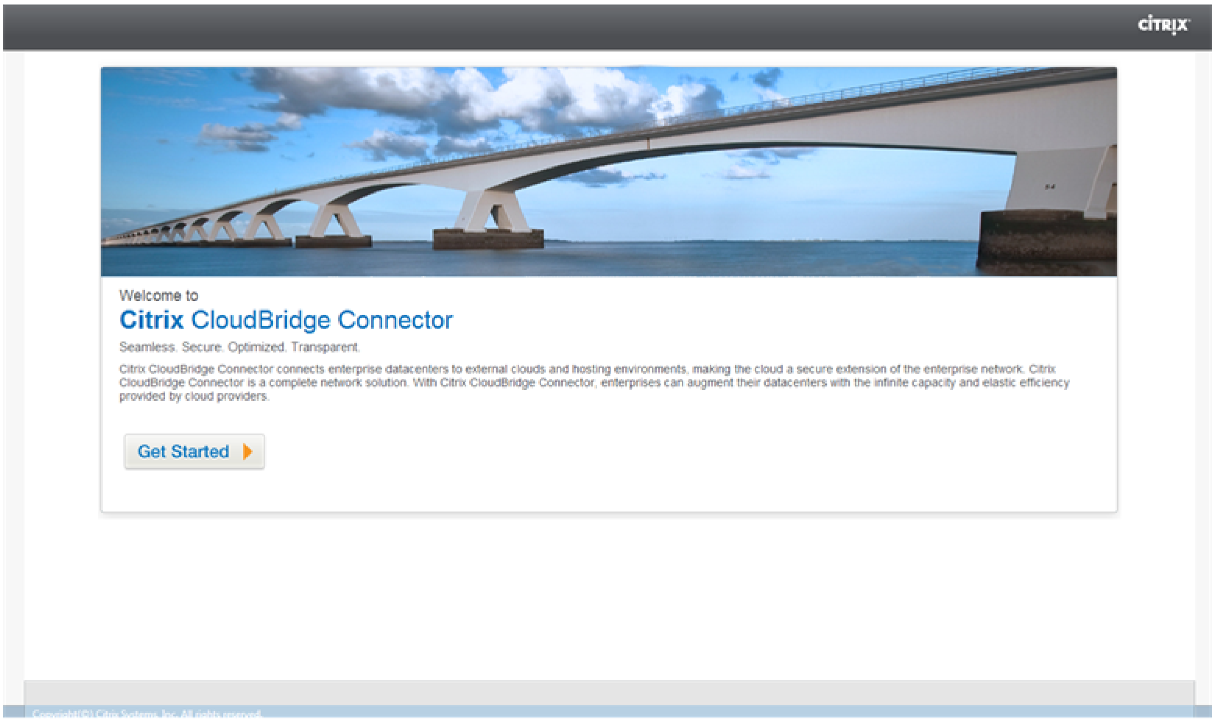
    Done
    > apply ns pbrs

    Done
<!--NeedCopy-->
```

To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the GUI

1. Type the NSIP address of a NetScaler appliance in the address line of a web browser.
2. Log on to the GUI of the NetScaler appliance by using your account credentials for the appliance.
3. Navigate to **System** > **CloudBridge Connector**.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.

   The first time you configure a CloudBridge Connector tunnel on the appliance, a **Welcome** screen appears.
5. On the **Welcome** screen click **Get Started**.


CITRIX
Get Started

Welcome to

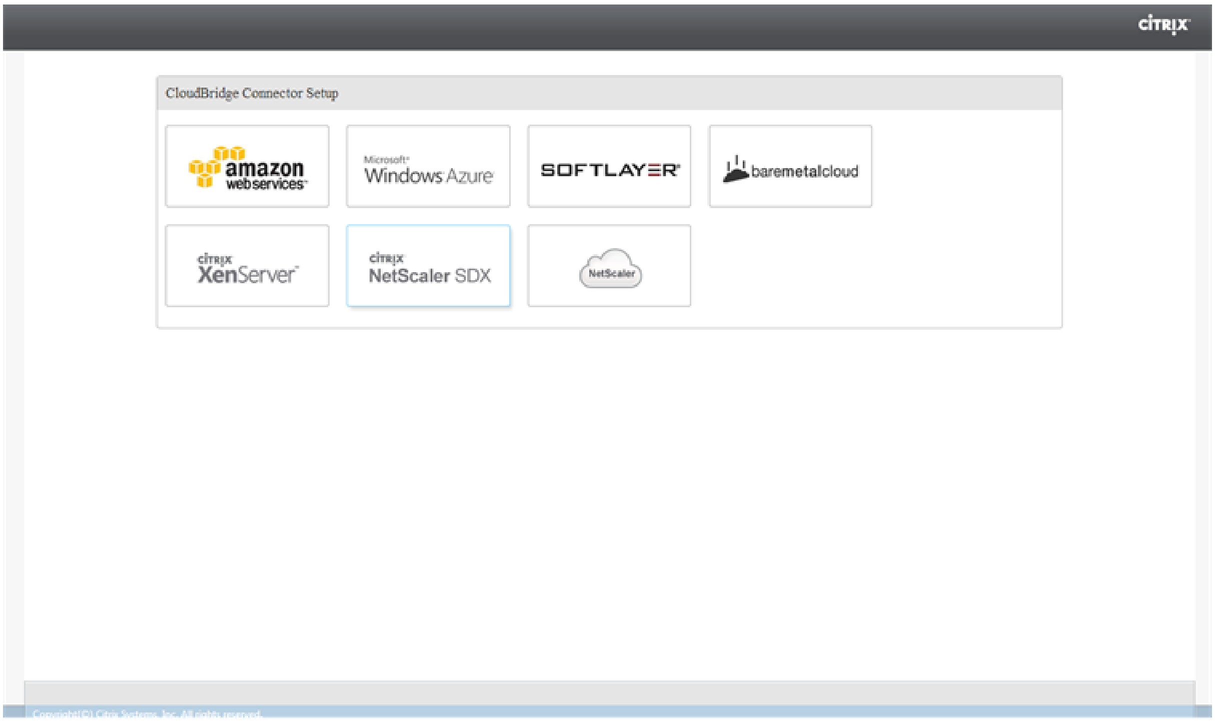
Citrix CloudBridge Connector

Seamless. Secure. Optimized. Transparent.

Citrix CloudBridge Connector connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of the enterprise network. Citrix CloudBridge Connector is a complete network solution. With Citrix CloudBridge Connector, enterprises can augment their datacenters with the infinite capacity and elastic efficiency provided by cloud providers.

> **Note**:
>
> If you already have a CloudBridge Connector tunnel configured on the NetScaler appliance, the Welcome screen does not appear, so you do not click Get Started.

1. In the **CloudBridge Connector Setup** pane, click **NetScaler**.


CITRIX
CloudBridge Connector Setup
amazon web services
Microsoft Windows Azure
SOFTLAYER
baremetalcloud
CITRIX XenServer
CITRIX NetScaler SDX
NetScaler

1. In the NetScaler pane, provide your account credentials for the remote NetScaler appliance. Click **Continue**.
2. In the **CloudBridge Connector Setting** pane, set the following parameter:
    - **CloudBridge Connector Name**—Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.
3. Under **Local Setting**, set the following parameter:
    - **Subnet IP**—IP address of the local endpoint of the CloudBridge Connector tunnel.
4. Under **Remote Setting**, set the following parameter:
    - **Subnet IP**—IP address of the peer endpoint of the CloudBridge Connector tunnel.
5. Under **PBR** Setting, set the following parameters:
    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - **Source IP Low** —Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Source IP High**—Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
    - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
    - Destination IP Low*—Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
    - **Destination IP High**—Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
6. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:
    - **Encryption Algorithm**—Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Hash Algorithm**—Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
    - **Key**—Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
        - **Auto Generate Key**—Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
        - **Specific Key**—Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.

        - Pre Shared Security Key—The text string entered for pre-shared key based authentication.
    - **Upload Certificates**—Authentication based on digital certificates.
        - **Public Key**—A local digital certificate to be used to authenticate the local NetScaler appliance to the peer before establishing IPSec security associations. The same certificate should be present and set for the Peer Public Key parameter in the peer.
        - **Private Key**—Private key of the local digital certificate.
        - **Peer Public Key**—Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.

7. Click **Done**.

The new CloudBridge Connector tunnel configuration on both the NetScaler appliances appears on the Home tab of the respective GUI. The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge Connectors pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can monitor the performance of CloudBridge Connector tunnels on a NetScaler appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see Monitoring CloudBridge Connector Tunnels.

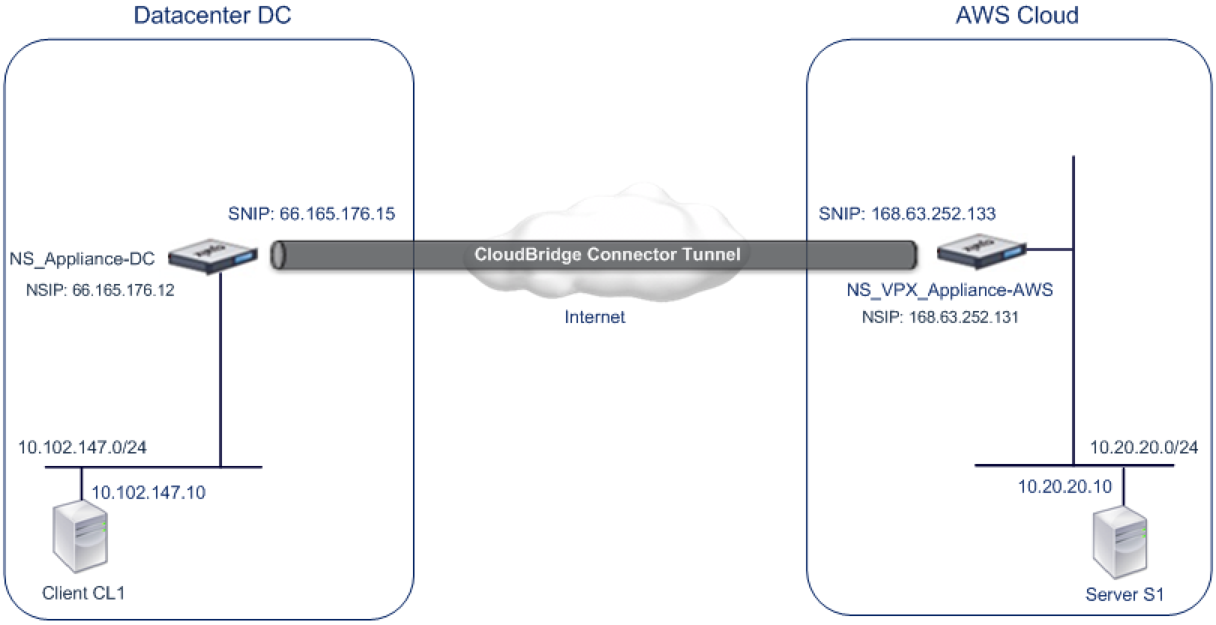
Configuring CloudBridge Connector between datacenter and AWS cloud

September 21, 2020

You can configure a CloudBridge Connector tunnel between a datacenter and AWS cloud to leverage the infrastructure and computing capabilities of the data center and the AWS cloud. With AWS, you can extend your network without initial capital investment or the cost of maintaining the extended network infrastructure. You can scale your infrastructure up or down, as required. For example, you can lease more server capabilities when the demand increases.

To connect a datacenter to AWS cloud, you set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in the datacenter and a NetScaler virtual appliance (VPX) that resides in AWS cloud.

As an illustration of a CloudBridge Connector tunnel between a datacenter and Amazon AWS cloud, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance NS_Appliance-DC, in datacenter DC, and NetScaler virtual appliance (VPX) NS_VPX_Appliance-AWS.


Datacenter DC
AWS Cloud
SNIP: 66.165.176.15
NS_Appliance-DC
NSIP: 66.165.176.12
CloudBridge Connector Tunnel
Internet
SNIP: 168.63.252.133
NS_VPX_Appliance-AWS
NSIP: 168.63.252.131
10.102.147.0/24
10.102.147.10
Client CL1
10.20.20.0/24
10.20.20.10
Server S1

Both NS_Appliance-DC and NS_VPX_Appliance-AWS function in L3 mode. They enable communication between private networks in datacenter DC and the AWS cloud. NS_Appliance-DC and NS_VPX_Appliance-AWS enable communication between client CL1 in datacenter DC and server S1 in the AWS cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

> **Note**:
>
> AWS does not support L2 mode, hence it is necessary to have only L3 mode enabled on both the endpoints.

For proper communication between CL1 and S1, L3 mode is enabled on NS_Appliance-DC and NS_VPX_Appliance-AWS and routes are updated as such:

- CL1 have a route to NS_Appliance-DC for reaching S1.
- NS_Appliance-DC have a route to NS_VPX_Appliance-AWS for reaching S1.
- S1 should have a route to NS_VPX_Appliance-AWS for reaching CL1.
- NS_VPX_Appliance-AWS have a route to NS_Appliance-DC for reaching CL1.

The following table lists the settings on NetScaler appliance NS_Appliance-DC in datacenter DC.

Entity	Name	Details
The NSIP address		66.165.176.12
SNIP address		66.165.176.15

Entity	Name	Details
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	Local endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15, Remote endpoint IP address of the CloudBridge Connector tunnel: 168.63.252.133, GRE Tunnel Details - Name= CC_Tunnel_DC-AWS

The following table lists the settings on NetScaler VPX NS_VPX_Appliance-AWS on AWS cloud.

Entity	Name	Details
NSIP address		10.102.25.30
Public EIP address mapped to the NSIP address		168.63.252.131
SNIP address		10.102.29.30
Public EIP address mapped to the SNIP address		168.63.252.133
CloudBridge Connector tunnel	CC_Tunnel_DC-AWS	Local endpoint IP address of the CloudBridge Connector tunnel:168.63.252.133, Remote endpoint IP address of the CloudBridge Connector tunnel: 66.165.176.15; **GRE Tunnel Details** Name= CC_Tunnel_DC-AWS, IPSec Profile Details, Name= CC_Tunnel_DC-AWS, Encryption algorithm= AES, Hash algorithm= HMAC SHA1

Prerequisites

Before setting up a CloudBridge Connector tunnel, verify that the following tasks have been completed:

1. Install, configure, and launch an instance of NetScaler Virtual appliance (VPX) on AWS cloud. For instructions on installing NetScaler VPX on AWS, see Deploy a NetScaler instance on AWS.
2. Deploy and configure a NetScaler physical appliance, or provisioning and configuring a NetScaler virtual appliance (VPX) on a virtualization platform in the datacenter.
3. Make sure that the CloudBridge Connector tunnel end-point IP addresses are accessible to each other.

NetScaler VPX License

After the initial instance launch, NetScaler VPX for AWS requires a license. If you are bringing your own license (BYOL), see the VPX Licensing Guide at: http://support.citrix.com/article/CTX122426.

You have to:

1. Use the licensing portal within Citrix website to generate a valid license.
2. Upload the license to the instance.

If this is a **paid** marketplace instance, then you do not need to install a license. The correct feature set and performance will activate automatically.

Configuration Steps

To set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in a datacenter and a NetScaler virtual appliance (VPX) that resides on the AWS cloud, use the GUI of the NetScaler appliance.

When you use the GUI, the CloudBridge Connector tunnel configuration created on the NetScaler appliance, is automatically pushed to the other endpoint or peer (the NetScaler VPX on AWS) of the CloudBridge Connector tunnel. Therefore, you do not have to access the GUI (GUI) of the NetScaler VPX on AWS to create the corresponding CloudBridge Connector tunnel configuration on it.

The CloudBridge Connector tunnel configuration on both peers (the NetScaler appliance that resides in the datacenter and the NetScaler virtual appliance (VPX) that resides on the AWS cloud) consists of the following entities:

- **IPSec profile**—An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in both the peers of the CloudBridge Connector tunnel.

- **GRE tunnel**—An IP tunnel specifies a local IP address (a public SNIP address configured on the local peer), remote IP address (a public SNIP address configured on the remote peer), protocol (GRE) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity.
- **Create a PBR rule and associate the IP tunnel with it**—A PBR entity specifies a set of conditions and an IP tunnel entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range and the destination IP address range to specify the subnet whose traffic is to traverse the CloudBridge Connector tunnel. For example, consider a request packet that originates from a client on the subnet in the datacenter and is destined to a server on the subnet in the AWS cloud. If this packet matches the source and destination IP address range of the PBR entity on the NetScaler appliance in the datacenter, it is sent across the CloudBridge Connector tunnel associated with the PBR entity.

To create an IPSEC profile by using the command line interface

At the command prompt, type:

- `add ipsec profile <name> [-**ikeVersion** (V1 | V2)] [-**encAlgo** (AES | 3DES)...] [-**hashAlgo** <hashAlgo> ...] [-**lifetime** <positive_integer>] (-**psk** | (-**publickey** <string> -**privatekey** <string> -**peerPublicKey** <string>))[-**livenessCheckInterval** <positive_integer>] [-**replayWindowSize** <positive_integer>] [-**ikeRetryInterval** <positive_integer>] [-**retransmissiontime** <positive_integer>]`
- `**show ipsec profile** <name>`

To create an IP tunnel and bind the IPSEC profile to it by using the command line interface

At the command prompt, type:

- `add ipTunnel <name> <remote><remoteSubnetMask> <local> [-protocol <protocol>] [-ipsecProfileName <string>]`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the command line interface

At the command prompt, type:

- `add ns pbr <pbr_name> ALLOW -srcIP = <local_subnet_range> -destIP = <remote_subnet_range> -ipTunnel <tunnel_name>`
- `apply ns pbrs`
- `show ns pbr <pbr_name>`

Example

```
    > add ipsec profile CC_Tunnel_DC-AWS -encAlgo AES -hashAlgo
       HMAC_SHA1

    Done
    > add ipTunnel CC_Tunnel_DC-AWS 168.63.252.133 255.255.255.0 66.165.176.15 -protocol GRE -ipsecProfileName CC_Tunnel_DC-AWS

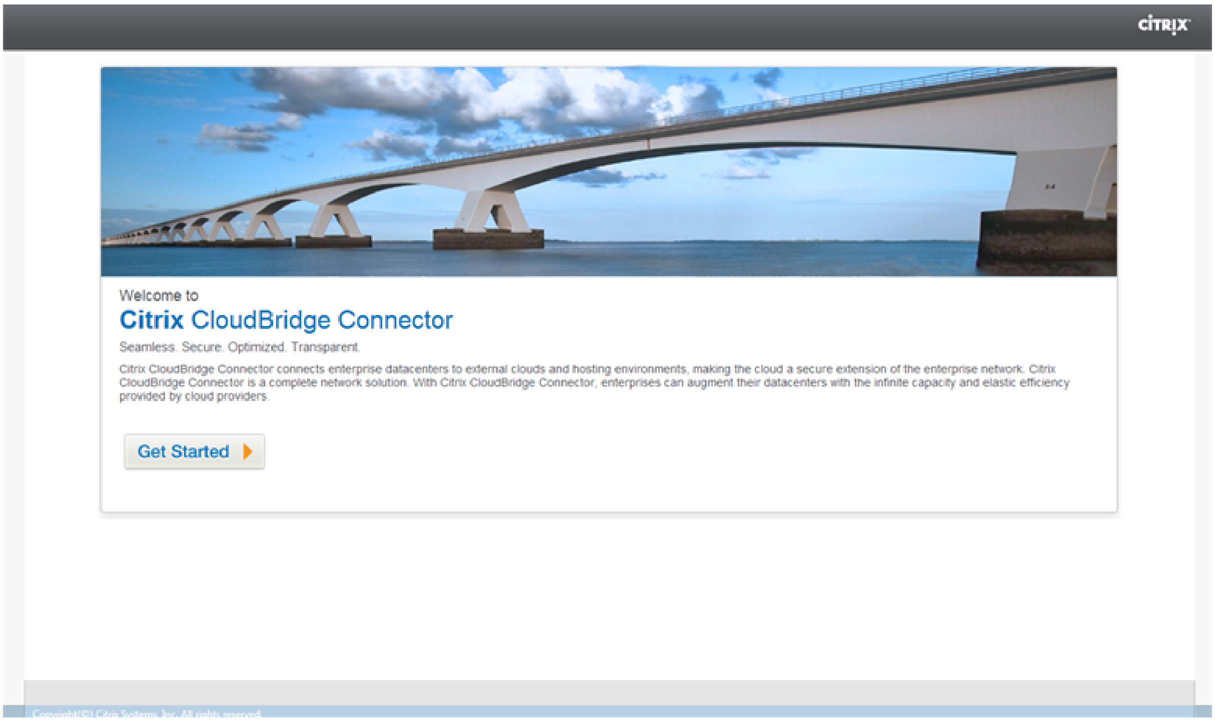
    Done
    > add ns pbr PBR-DC-AWS ALLOW -srcIP 66.165.176.15 -destIP 168.63.252.133 ipTunnel CC_Tunnel_DC-AWS

    Done
    > apply ns pbrs

    Done
<!--NeedCopy-->
```

To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the GUI

1. Type the NSIP address of a NetScaler appliance in the address line of a web browser.
2. Log on to the GUI of the NetScaler appliance by using your account credentials for the appliance.
3. Navigate to **System** > **CloudBridge Connector**.
4. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge**.
5. The first time you configure a CloudBridge Connector tunnel on the appliance, a **Welcome** screen appears.
6. On the **Welcome** screen click **Get Started**.



Welcome to

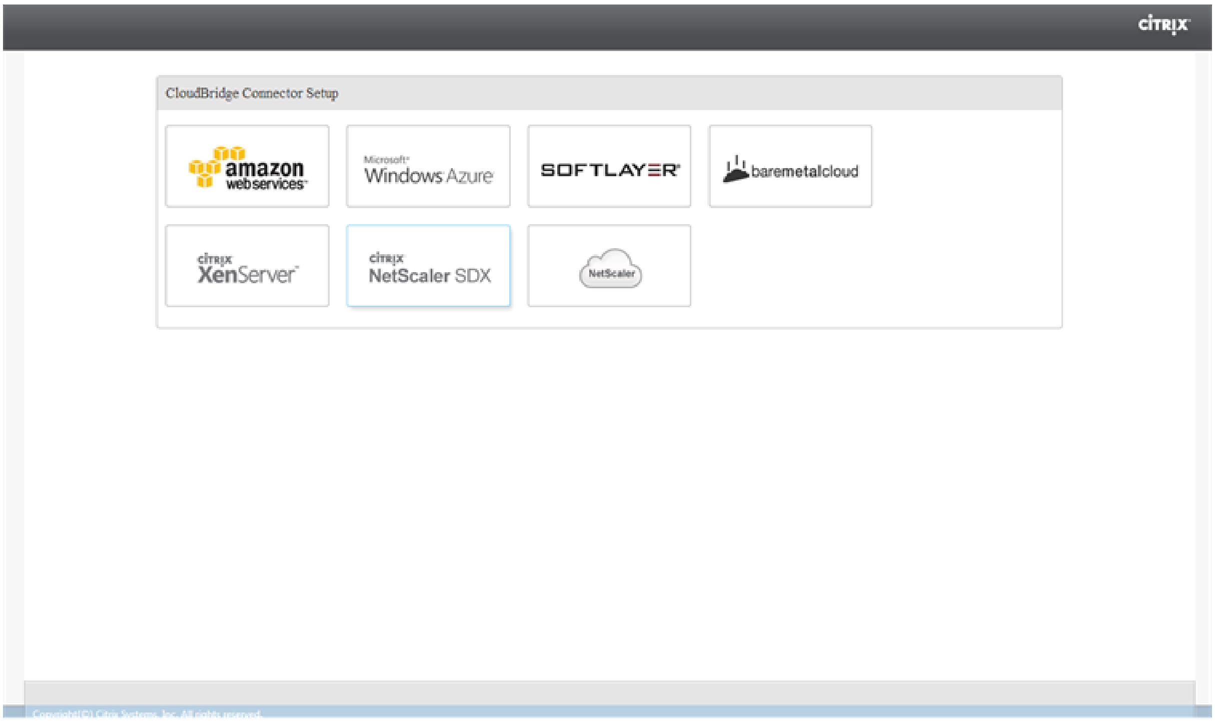
Citrix CloudBridge Connector

Seamless. Secure. Optimized. Transparent.

Citrix CloudBridge Connector connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of the enterprise network. Citrix CloudBridge Connector is a complete network solution. With Citrix CloudBridge Connector, enterprises can augment their datacenters with the infinite capacity and elastic efficiency provided by cloud providers.

> **Note**:
>
> If you already have a CloudBridge Connector tunnel configured on the NetScaler appliance, the Welcome screen does not appear, so you do not click Get Started.

1. In the **CloudBridge Connector Setup** pane, click **amazon web services**


CloudBridge Connector Setup
amazon web services
Microsoft Windows Azure
SOFTLAYER
baremetalcloud
citrix XenServer
citrix NetScaler SDX
NetScaler
Copyright(C) Citrix Systems, Inc. All rights reserved.

1. In the **Amazon** pane, provide your AWS account credentials: AWS Access Key ID and AWS Secret Access Key. You can obtain these access keys from the AWS GUI console. Click **Continue**.

> **Note**
>
> Earlier, the Setup wizard always connects to the same AWS region even when another region is selected. As a result, configuring CloudBridge Connector tunnel to a NetScaler VPX running on the selected AWS region used to fail. This issue has been fixed now.

1. In the **NetScaler** pane, select the NSIP address of the NetScaler virtual appliance running on AWS. Then, provide your account credentials for the NetScaler virtual appliance. Click **Continue**.
2. In the **CloudBridge Connector Setting** pane, set the following parameter:
    - **CloudBridge Connector Name**—Name for the CloudBridge Connector configuration on the local appliance. Must begin with an ASCII alphabetic or underscore (_) character, and must contain only ASCII alphanumeric, underscore, hash (#), period (.), space, colon (:), at (@), equals (=), and hyphen (-) characters. Cannot be changed after the CloudBridge Connector configuration is created.

3. Under **Local Setting**, set the following parameter:
   - **Subnet IP**—IP address of the local endpoint of the CloudBridge Connector tunnel. Must be a public IP address of type SNIP.
4. Under **Remote Setting**, set the following parameter:
   - **Subnet IP**—IP address of the CloudBridge Connector tunnel end point on the AWS side. Must be an IP address of type SNIP on the NetScaler VPX instance on AWS.
   - **NAT**—Public IP address (EIP) in AWS that is mapped to the SNIP configured on the NetScaler VPX instance on AWS.
5. Under **PBR Setting**, set the following parameters:
   - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
   - **Source IP Low**—Lowest source IP address to match against the source IP address of an outgoing IPv4 packet.
   - **Source IP High**—Highest source IP address to match against the source IP address of an outgoing IPv4 packet.
   - **Operation**—Either the equals (=) or does not equal (!=) logical operator.
   - **Destination IP Low**—Lowest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
   - **Destination IP High**—Highest destination IP address to match against the destination IP address of an outgoing IPv4 packet.
6. (Optional) Under **Security Settings**, set the following IPSec protocol parameters for the CloudBridge Connector tunnel:
   - **Encryption Algorithm**—Encryption algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
   - **Hash Algorithm**—Hash algorithm to be used by the IPSec protocol in the CloudBridge tunnel.
   - **Key**— Select one of the following IPSec authentication methods to be used by the two peers to mutually authenticate.
     - **Auto Generate Key**— Authentication based on a text string, called a pre-shared key (PSK), generated automatically by the local appliance. The PSKs keys of the peers are matched against each other for authentication.
     - **Specific Key**—Authentication based on a manually entered PSK. The PSKs of the peers are matched against each other for authentication.
       - **Pre Shared Security Key**—The text string entered for pre-shared key based authentication.
     - **Upload Certificates**—Authentication based on digital certificates.
       - **Public Key**—A local digital certificate to be used to authenticate the local peer to the remote peer before establishing IPSec security associations. The same certifi-

cate should be present and set for the Peer Public Key parameter in the peer.

- **Private Key**—Private key of the local digital certificate.
- **Peer Public Key**—Digital certificate of the peer. Used to authenticate the peer to the local end point before establishing IPSec security associations. The same certificate should be present and set for the Public key parameter in the peer.

7. Click **Done**.

The new CloudBridge Connector tunnel configuration on the NetScaler appliance in the datacenter appears on the Home tab of the GUI. The corresponding new CloudBridge Connector tunnel configuration on the NetScaler VPX appliance in the AWS cloud appears on the GUI. The current status of the CloudBridge connector tunnel is indicated in the Configured CloudBridge pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a NetScaler appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see Monitoring CloudBridge Connector Tunnels.

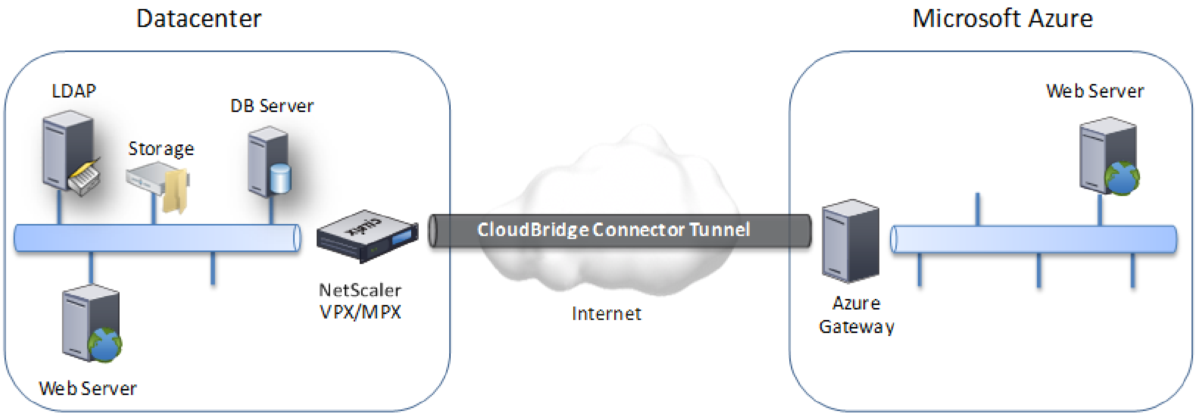
Configuring a CloudBridge Connector tunnel between a datacenter and Azure cloud

September 21, 2020

The NetScaler appliance provides connectivity between your enterprise datacenters and the Microsoft cloud hosting provider, Azure, making Azure a seamless extension of the enterprise network. NetScaler encrypts the connection between the enterprise datacenter and Azure cloud so that all data transferred between the two is secure.

How CloudBridge Connector tunnel works

To connect a datacenter to Azure cloud, you set up a CloudBridge Connector tunnel between a NetScaler appliance that resides in the datacenter and a gateway that resides in the Azure cloud. The NetScaler appliance in the datacenter and the gateway in Azure cloud are the end points of the CloudBridge Connector tunnel and are called peers of the CloudBridge Connector tunnel.



A CloudBridge Connector tunnel between a datacenter and Azure cloud uses the open-standard Internet Protocol security (IPSec) protocol suite, in tunnel mode, to secure communications between peers in the CloudBridge Connector tunnel. In a CloudBridge Connector tunnel, IPSec ensures:

- Data integrity
- Data origin authentication
- Data confidentiality (encryption)
- Protection against replay attacks

IPSec uses the tunnel mode in which the complete IP packet is encrypted and then encapsulated. The encryption uses the Encapsulating Security Payload (ESP) protocol, which ensures the integrity of the packet by using a HMAC hash function and ensures confidentiality by using an encryption algorithm. The ESP protocol, after encrypting the payload and calculating the HMAC, generates an ESP header and inserts it before the encrypted IP packet. The ESP protocol also generates an ESP trailer and inserts it at the end of the packet.

The IPSec protocol then encapsulates the resulting packet by adding an IP header before the ESP header. In the IP header, the destination IP address is set to the IP address of the CloudBridge Connecter peer.

Peers in the CloudBridge Connector tunnel use the Internet Key Exchange version 1 (IKEv1) protocol (part of the IPSec protocol suite) to negotiate secure communication, as follows:

1. The two peers mutually authenticate with each other, using pre-shared key authentication, in which the peers exchange a text string called a pre-shared key (PSK). The pre-shared keys are matched against each other for authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on each of the peers.
2. The peers then negotiate to reach agreement on:
    - An encryption algorithm
    - Cryptographic keys for encrypting data on one peer and decrypting it on the other.

   This agreement upon the security protocol, encryption algorithm and cryptographic keys is called a Security Association (SA). SAs are one-way (simplex). For example, when a CloudBridge

> Connector tunnel is set up between a NetScaler appliance in a datacenter and a gateway in an Azure cloud, both the datacenter appliance and the Azure gateway have two SAs. One SA is used for processing out-bound packets, and the other SA is used for processing inbound packets. SAs expire after a specified interval of time, which is called the lifetime.

Example of CloudBridge Connector tunnel configuration and data flow

As an illustration of CloudBridge Connector Tunnel, consider an example in which a CloudBridge Connector tunnel is set up between NetScaler appliance CB_Appliance-1 in a datacenter and gateway Azure_Gateway-1 in Azure cloud.

CB_Appliance-1 also functions as an L3 router, which enables a private network in the datacenter to reach a private network in the Azure cloud through the CloudBridge Connector tunnel. As a router, CB_Appliance-1 enables communication between client CL1 in the datacenter and server S1 in the Azure cloud through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On CB_Appliance-1, the CloudBridge Connector tunnel configuration includes an IPSec profile entity named CB_Azure_IPSec_Profile, a CloudBridge Connector tunnel entity named CB_Azure_Tunnel, and a policy based routing (PBR) entity named CB_Azure_Pbr.

The IPSec profile entity CB_Azure_IPSec_Profile specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, and hash algorithm, to be used by the IPSec protocol in the CloudBridge Connector tunnel. CB_Azure_IPSec_Profile is bound to IP tunnel entity CB_Azure_Tunnel.

CloudBridge Connector tunnel entity CB_Azure_Tunnel specifies the local IP address (a public IP (SNIP) address configured on the NetScaler appliance), the remote IP address (the IP address of the Azure_Gateway-1), and the protocol (IPSec) used to set up the CloudBridge Connector tunnel. CB_Azure_Tunnel is bound to the PBR entity CB_Azure_Pbr.

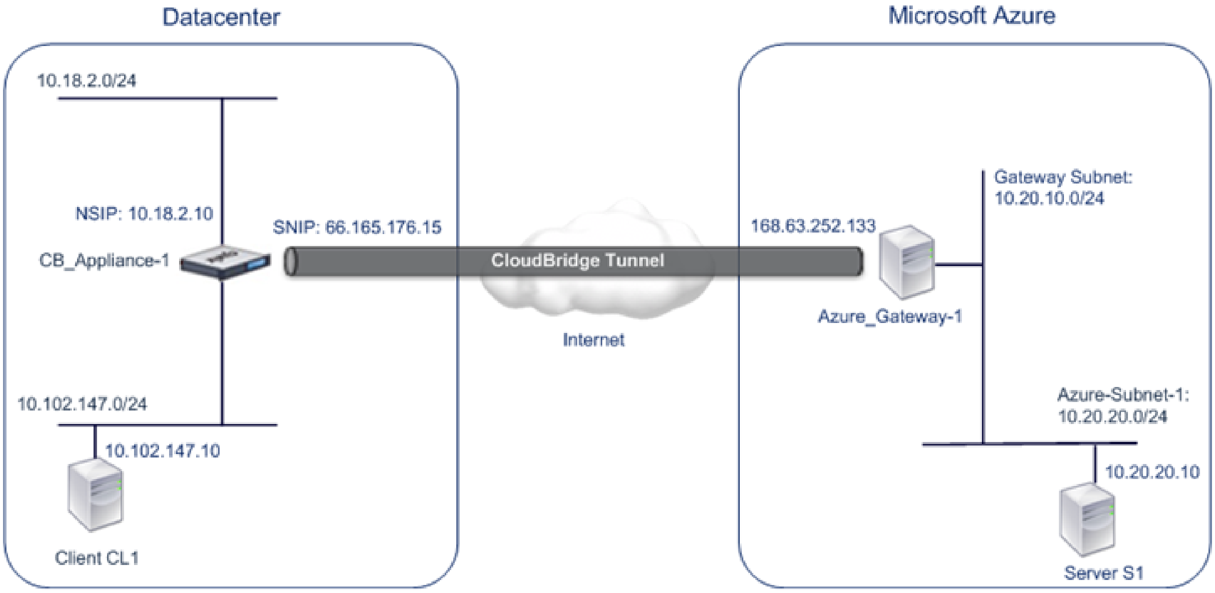
The PBR entity CB_Azure_Pbr specifies a set of conditions and a CloudBridge Connector tunnel entity (CB_Azure_Tunnel). The source IP address range and the destination IP address range are the conditions for CB_Azure_Pbr. The source IP address range and the destination IP address range are specified as a subnet in the datacenter and a subnet in the Azure cloud, respectively. Any request packet originating from a client in the subnet in the datacenter and destined to a server in the subnet on the Azure cloud matches the conditions in CB_Azure_Pbr. This packet is then considered for CloudBridge processing and is sent across the CloudBridge Connector tunnel (CB_Azure_Tunnel) bound to the PBR entity.

On Microsoft Azure, the CloudBridge Connector tunnel configuration includes a local network entity named My-Datacenter-Network, a virtual network entity named Azure-Network-for-CloudBridge-Tunnel, and a gateway named Azure_Gateway-1.

The local (local to Azure) network entity My-Datacenter-Network specifies the IP address of the

NetScaler appliance on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network entity Azure-Network-for-CloudBridge-Tunnel defines a private subnet named Azure-Subnet-1 in Azure. The traffic of the subnet traverses the CloudBridge Connector tunnel. The server S1 is provisioned in this subnet.

The local network entity My-Datacenter-Network is associated with the virtual network entity Azure-Network-for-CloudBridge-Tunnel. This association defines the remote and local network details of the CloudBridge Connector tunnel configuration in Azure. Gateway Azure_Gateway-1 was created for this association to become the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel.


Datacenter
Microsoft Azure
10.18.2.0/24
NSIP: 10.18.2.10
CB_Appliance-1
SNIP: 66.165.176.15
CloudBridge Tunnel
Internet
168.63.252.133
Gateway Subnet:
10.20.10.0/24
Azure_Gateway-1
10.102.147.0/24
10.102.147.10
Client CL1
Azure-Subnet-1:
10.20.20.0/24
10.20.20.10
Server S1

For more information about the settings, refer to the CloudBridge Connector Tunnel Settings pdf.

Points to consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a NetScaler appliance in datacenter and Microsoft Azure, consider the following points:

1. The NetScaler appliance must have a public facing IPv4 address (type SNIP) to use as a tunnel end-point address for the CloudBridge Connector tunnel. Also, the NetScaler appliance should not be behind a NAT device.
2. Azure supports the following IPSec settings for a CloudBridge Connector tunnel. Therefore, you must specify the same IPSec settings while configuring the NetScaler for the CloudBridge Connector tunnel.
    - IKE version = v1
    - Encryption algorithm = AES
    - Hash algorithm = HMAC SHA1
3. You must configure the firewall in the datacenter edge to allow the following.

    - Any UDP packets for port 500
    - Any UDP packets for port 4500
    - Any ESP (IP protocol number 50) packets
4. IKE re-keying, which is renegotiation of new cryptographic keys between the CloudBridge Connector tunnel end points to establish new SAs, is not supported. When the Security Associations (SAs) expire, the tunnel goes into the DOWN state. Therefore, you must set a very large value for the lifetimes of SAs.
5. You must configure Microsoft Azure before specifying the tunnel configuration on the NetScaler, because the public IP address of the Azure end (gateway) of the tunnel, and the PSK, are automatically generated when you set up the tunnel configuration in Azure. You need this information for specifying the tunnel configuration on the NetScaler.

Configuring the CloudBridge Connector tunnel

For setting up a CloudBridge Connector tunnel between your datacenter and Azure, you must install CloudBridge VPX/MPX in your datacenter, configure Microsoft Azure for the CloudBridge Connector tunnel, and then configure the NetScaler appliance in the data center for the CloudBridge Connector tunnel.

Configuring a CloudBridge Connector tunnel between a NetScaler appliance in datacenter and Microsoft Azure consists of the following tasks:

1. **Setting up the NetScaler appliance in the datacenter**. This task involves deploying and configuring a NetScaler physical appliance (MPX), or provisioning and configuring a NetScaler virtual appliance (VPX) on a virtualization platform in the datacenter.
2. **Configuring Microsoft Azure for the CloudBridge Connector tunnel**. This task involves creating local network, virtual network, and gateway entities in Azure. The local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the NetScaler appliance) on the datacenter side, and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel. The virtual network defines a network on Azure. Creating the virtual network includes defining a subnet whose traffic is to traverse the CloudBridge Connector tunnel to be formed. You then associate the local network with the virtual network. Finally, you create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel.
3. **Configuring the NetScaler appliance in the datacenter for the CloudBridge Connector tunnel**. This task involves creating an IPSec profile, an IP tunnel entity, and a PBR entity in the NetScaler appliance in datacenter. The IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used in the CloudBridge Connector tunnel. The IP tunnel specifies the IP address of both the CloudBridge Connector tunnel end points (the NetScaler appliance in datacenter and the gateway in Azure) and the protocol to be used in the CloudBridge Connector tunnel. You then associate the IPSec profile entity with the IP tunnel entity. The PBR entity specifies the two subnets, in the datacen-

ter and in the Azure cloud, that are to communicate with each other through the CloudBridge Connector tunnel. You then associate the IP tunnel entity with the PBR entity.

Configuring Microsoft Azure for the CloudBridge Connector tunnel

To create a CloudBridge Connector tunnel configuration on Microsoft Azure, use the Microsoft Windows Azure Management Portal, which is a web based graphical interface for creating and managing resources on Microsoft Azure.

Before you begin the CloudBridge Connector tunnel configuration on Azure cloud, make sure that:

- You have a user account for Microsoft Azure.
- You have a conceptual understanding of Microsoft Azure.
- You are familiar with the Microsoft Windows Azure Management Portal.

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on Microsoft Azure by using the Microsoft Windows Azure Management Portal:

- **Create a local network entity**. Create a local network entity in Windows Azure for specifying the network details of the datacenter. A local network entity specifies the IP address of the CloudBridge Connector tunnel end point (the NetScaler) on the datacenter side and the datacenter subnet whose traffic is to traverse the CloudBridge Connector tunnel.
- **Create a Virtual Network**. Create virtual network entity that defines a network on Azure. This task includes defining a private address space, where you provide a range of private addresses and subnets belonging to the range specified in the address space. The traffic of the subnets will traverse the CloudBridge Connector tunnel. You then associate a local network entity with the virtual network entity. This association lets Azure create a configuration for a CloudBridge Connector tunnel between the virtual network and the data center network. A gateway (to be created) in Azure for this virtual network will be the CloudBridge end point at the Azure end of the CloudBridge Connector tunnel. You then define a private subnet for the gateway to be created. This subnet belongs to the range specified in the address space in the virtual network entity.
- **Create a gateway in Windows Azure**. Create a gateway that becomes the end point at the Azure end of the CloudBridge Connector tunnel. Azure, from its pool of public IP addresses, assigns an IP address to the gateway created.
- **Gather the public IP address of the gateway and the pre-shared key.** For a CloudBridge Connector tunnel configuration on Azure, the public IP address of the gateway and the pre-shared Key (PSK) are automatically generated by Azure. Make a note of this information. You will need it for configuring the CloudBridge Connector tunnel on the NetScaler in datacenter.

> **Note:**
>
> The procedures for configuring Microsoft Azure for a CloudBridge Connector tunnel might change

> over time, depending on the Microsoft Azure release cycle. For the latest procedures, see the Microsoft Azure documentation.

Configuring the NetScaler Appliance in the datacenter for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a datacenter and an Azure cloud, perform the following tasks on the NetScaler in the datacenter. You can use either the NetScaler command line or the GUI:

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and PSK, to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel with IPSec protocol and associate the IPSec profile to it**. An IP tunnel specifies the local IP address (a public SNIP address configured on the NetScaler appliance), remote IP address (the public IP address of the gateway in Azure), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate the IP tunnel to it**. A PBR entity specifies a set of conditions and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP range are the conditions for the PBR entity. You must set the source IP address range to specify the datacenter subnet whose traffic is to traverse the tunnel, and the destination IP address range to specify the Azure subnet whose traffic is to traverse the CloudBridge Connector tunnel. Any request packet originated from a client in the subnet on the datacenter and destined to a server in the subnet on the Azure cloud matches the source and destination IP range of the PBR entity. This packet is then considered for CloudBridge Connector tunnel processing and is sent across sent across the CloudBridge Connector tunnel associated with the PBR entity.

The GUI combines all these tasks in a single wizard called the CloudBridge Connector wizard.
To create an IPSEC profile by using the NetScaler command line:

At the Command prompt, type:

```
add ipsec profile <name> -psk <string> -ikeVersion v1
```

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line:

At the Command prompt, type:

```
add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -
ipsecProfileName <string>
```

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

```
add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> ipTunnel
 <tunnelName> apply pbrs
```

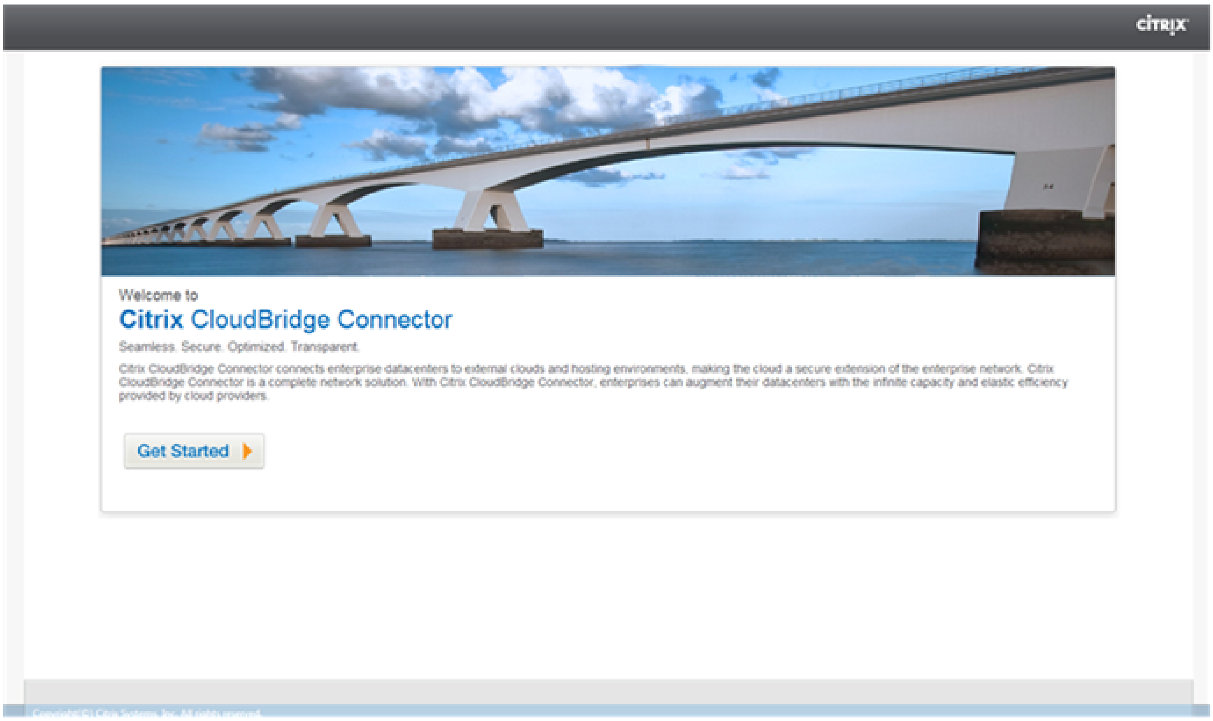
Sample Configuration

The following commands create all settings of NetScaler appliance CB_Appliance-1 used in “Example of CloudBridge Connector Configuration and Data Flow”.

```
   add ipsec profile CB_Azure_IPSec_Profile -psk DkiMgMdcbqvYREEuIvxsbKkW0FOyDiLM  -ikeVersion v1  - lifetime 31536000
    Done
    >  add iptunnel CB_Azure_Tunnel 168.63.252.133 255.255.255.255 66.165.176.15  - protocol IPSEC  - ipsecProfileName CB_Azure_IPSec_Profile
     Done
    > add pbr CB_Azure_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP 10.20.0.0-10.20.255.255  - ipTunnelCB_Azure_Tunnel
     Done
    > apply pbrs
    Done
<!--NeedCopy-->
```

To configure a CloudBridge Connector tunnel in a NetScaler appliance by using the GUI

1. Access the GUI by using a web browser to connect to the IP address of the NetScaler appliance in the datacenter.
2. Navigate to **System** > **CloudBridge Connector**.
3. In the right pane, under G**etting Started**, click **Create/Monitor CloudBridge**.
4. Click **Get Started**.



Welcome to

Citrix CloudBridge Connector

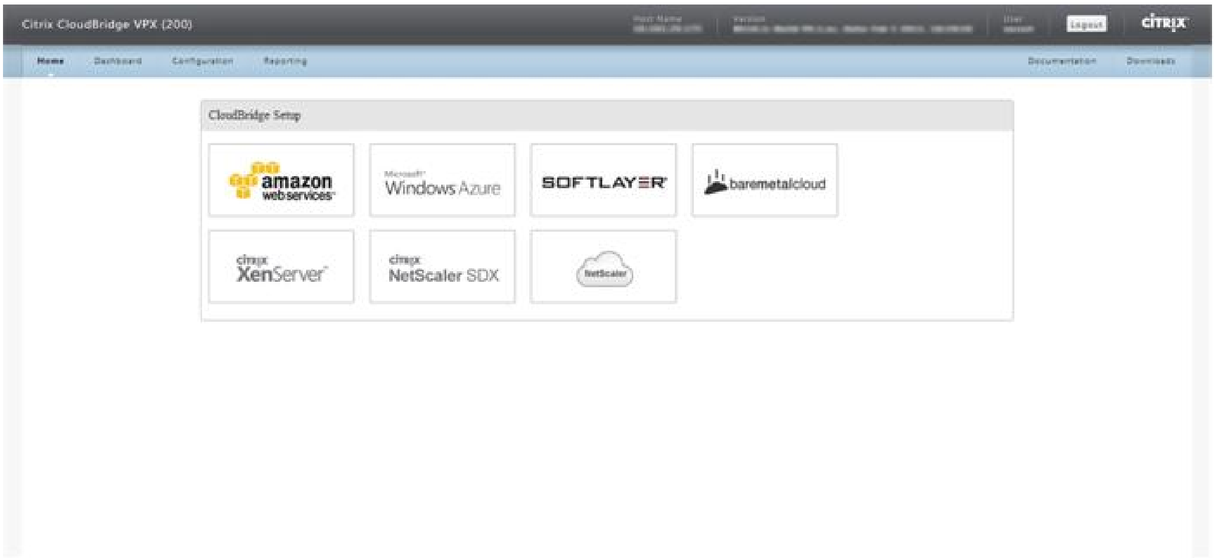
Seamless. Secure. Optimized. Transparent.

Citrix CloudBridge Connector connects enterprise datacenters to external clouds and hosting environments, making the cloud a secure extension of the enterprise network. Citrix CloudBridge Connector is a complete network solution. With Citrix CloudBridge Connector, enterprises can augment their datacenters with the infinite capacity and elastic efficiency provided by cloud providers.

Get Started

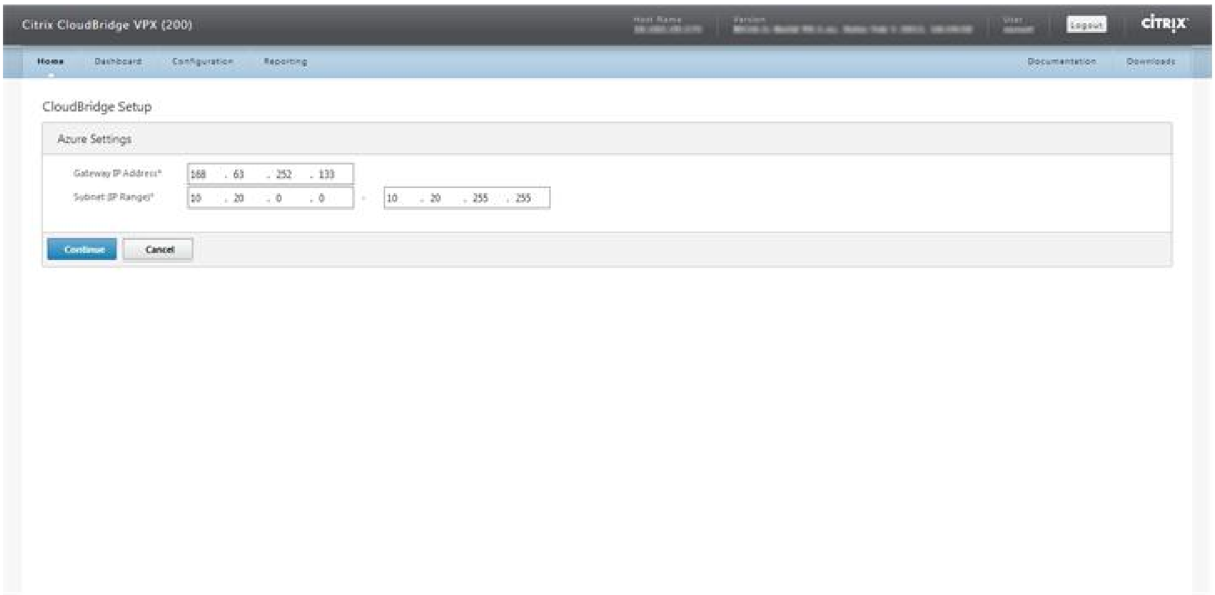
> **Note**: If you already have any CloudBridge Connector tunnel configured on the NetScaler appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.

5. In the CloudBridge Setup pane, click **Microsoft Windows Azure**.



Citrix CloudBridge VPX (200)
Logout
Home
Dashboard
Configuration
Reporting
Documentation
Downloads
CloudBridge Setup
amazon web services
Microsoft Windows Azure
SOFTLAYER
baremetalcloud
citrix XenServer
citrix NetScaler SDX
NetScaler

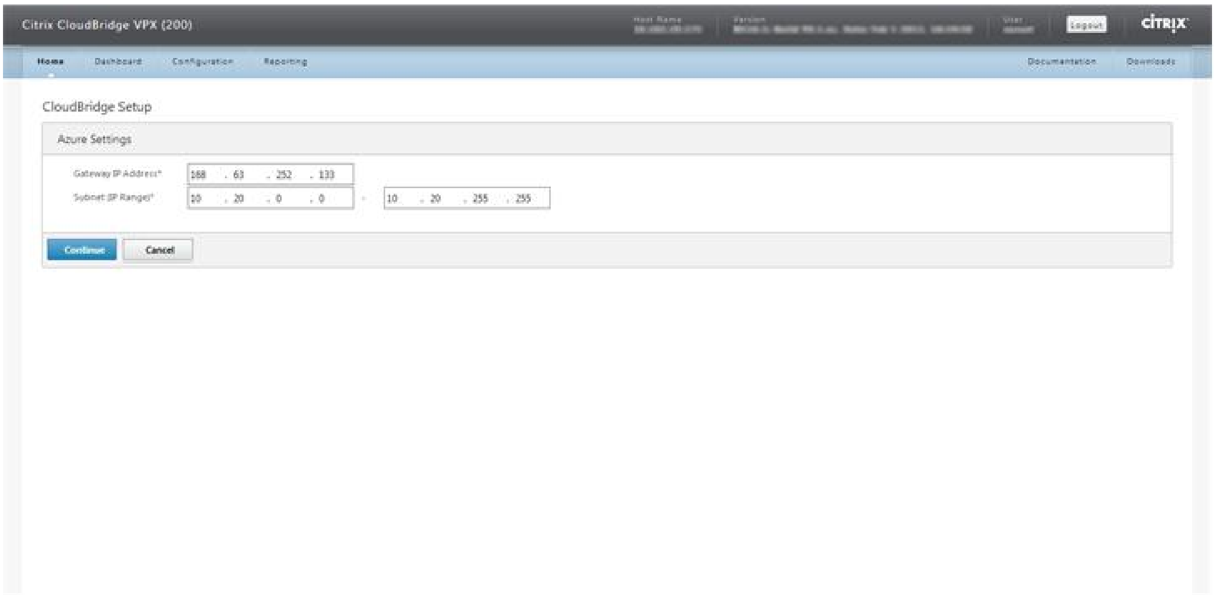
1. In the Azure Settings pane, in the **Gateway IP Address** field, type the IP address of the Azure gateway. The CloudBridge Connector tunnel is then set up between the NetScaler appliance and the gateway. In the **Subnet (IP Range)** text boxes, specify a subnet range (in Azure cloud), the traffic of which is to traverse the CloudBridge Connector tunnel. Click **Continue**.


Citrix CloudBridge VPX (200)
Logout
CITRIX
Home
Dashboard
Configuration
Reporting
Documentation
Downloads
CloudBridge Setup
Azure Settings
Continue
Cancel

Gateway IP Address* 168 . 63 . 252 . 133

Subnet (IP Range)* 10 . 20 . 0 . 0 - 10 . 20 . 255 . 255

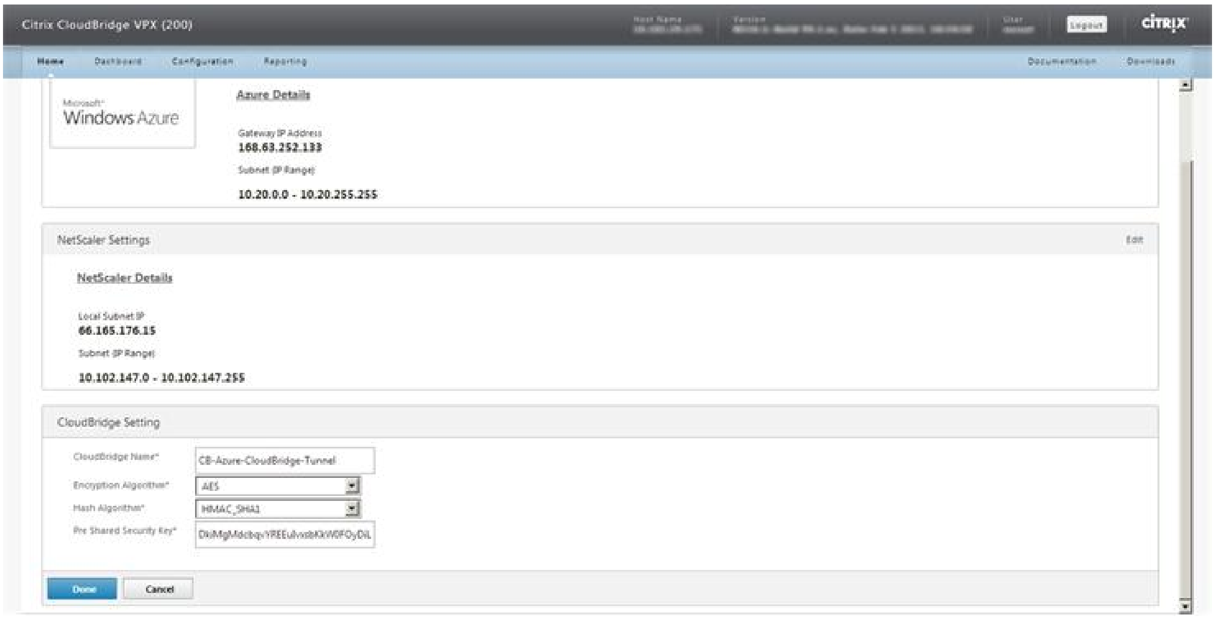
1. In the NetScaler Settings pane, from the **Local Subnet IP** drop-down list, select a publicly accessible SNIP address configured on the NetScaler appliance. In Subnet (IP Range)* text boxes, specify a local subnet range, the traffic of which is to traverse the CloudBridge Connector tunnel. Click **Continue**.


Citrix CloudBridge VPX (200)
Logout
CITRIX
Home
Dashboard
Configuration
Reporting
Documentation
Downloads
CloudBridge Setup
Azure Settings
Continue
Cancel

Gateway IP Address* 168 . 63 . 252 . 133

Subnet (IP Range)* 10 . 20 . 0 . 0 - 10 . 20 . 255 . 255

1. In the **CloudBridge Setting** pane, in the CloudBridge Name text box, type a name for the CloudBridge that you want to create.



Azure Details

Gateway IP Address
168.63.252.133
Subnet (IP Range)
10.20.0.0 - 10.20.255.255

NetScaler Settings

NetScaler Details

Local Subnet IP
66.165.176.15
Subnet (IP Range)
10.102.147.0 - 10.102.147.255

CloudBridge Setting

CloudBridge Name*: CB-Azure-CloudBridge-Tunnel
Encryption Algorithm*: AES
Hash Algorithm*: HMAC_SHA1
Pre Shared Security Key*

1. From the Encryption Algorithm and Hash Algorithm drop-down lists, select the AES and HMAC_SHA1 algorithms, respectively. In the Pre Shared Security Key text box, type the security key.
2. Click **Done**.

Monitoring the CloudBridge Connector tunnel

You can view statistics for monitoring the performance of a CloudBridge Connector tunnel between the NetScaler appliance in the datacenter and Microsoft Azure. To view CloudBridge Connector tunnel statistics on the NetScaler appliance, use GUI or NetScaler command line. To view CloudBridge Connector tunnel statistics in Microsoft Azure, use the Microsoft Windows Azure Management Portal.

Displaying CloudBridge Connector tunnel Statistics in the NetScaler appliance

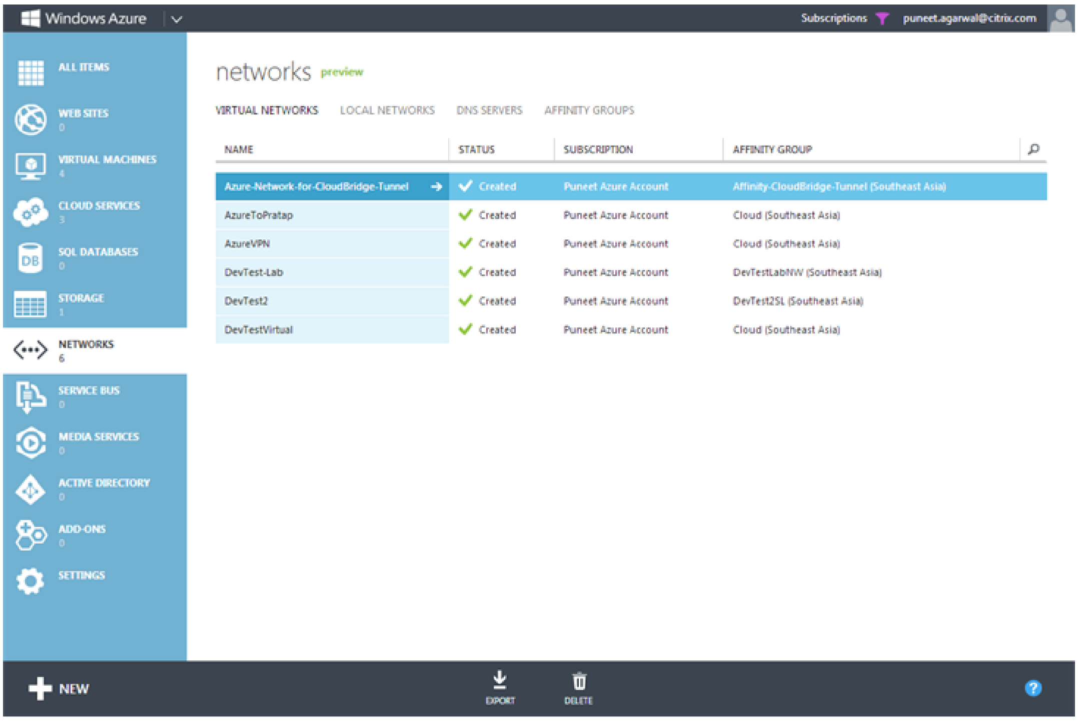
For information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see
Monitoring CloudBridge Connector Tunnels.

- Displaying CloudBridge Connector tunnel Statistics in Microsoft Azure

The following table lists the statistical counters available for monitoring CloudBridge Connector tunnels in Microsoft Azure.

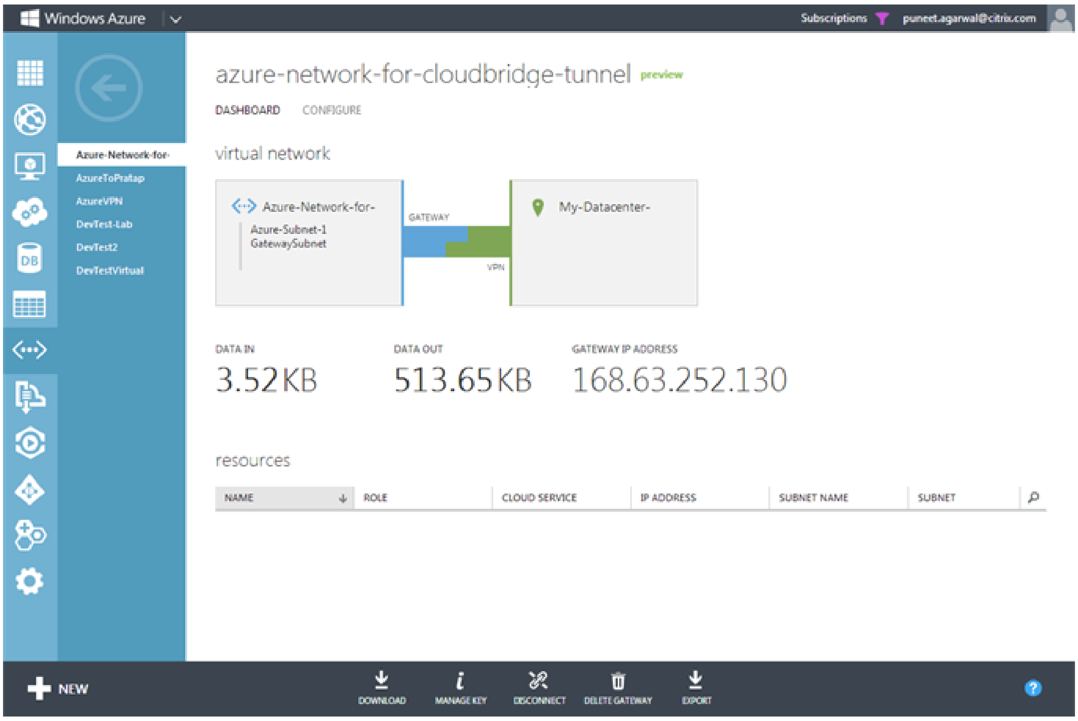
Statistical counter	Specifies
DATA IN	Total number of kilobytes received by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.
DATA OUT	Total number of kilobytes sent by the Azure gateway through the CloudBridge Connector tunnel since the gateway was created.

- To display CloudBridge Connector tunnel statistics by using the Microsoft Windows Azure Management Portal
    1. Log on to the Windows Azure Management Portal by using your Microsoft Azure account credentials.
    2. In the left pane, click **NETWORKS**.
    3. On the **Virtual Network** tab, in the Name column, select the virtual network entity associated with a CloudBridge Connector tunnel whose statistics you want to display.


Windows Azure
Subscriptions
puneet.agarwal@citrix.com
ALL ITEMS
WEB SITES
0
VIRTUAL MACHINES
4
CLOUD SERVICES
3
SQL DATABASES
0
STORAGE
1
NETWORKS
6
SERVICE BUS
0
MEDIA SERVICES
0
ACTIVE DIRECTORY
0
ADD-ONS
0
SETTINGS
networks preview
VIRTUAL NETWORKS
LOCAL NETWORKS
DNS SERVERS
AFFINITY GROUPS
NEW
EXPORT
DELETE

NAME	STATUS	SUBSCRIPTION	AFFINITY GROUP
Azure-Network-for-CloudBridge-Tunnel	✓ Created	Puneet Azure Account	Affinity-CloudBridge-Tunnel (Southeast Asia)
AzureToPratap	✓ Created	Puneet Azure Account	Cloud (Southeast Asia)
AzureVPN	✓ Created	Puneet Azure Account	Cloud (Southeast Asia)
DevTest-Lab	✓ Created	Puneet Azure Account	DevTestLabNW (Southeast Asia)
DevTest2	✓ Created	Puneet Azure Account	DevTest2SL (Southeast Asia)
DevTestVirtual	✓ Created	Puneet Azure Account	Cloud (Southeast Asia)

    4. On the **DASHBOARD** page of the virtual network, view the DATA IN and DATA OUT counters for the CloudBridge Connector tunnel.


Windows Azure
Subscriptions
puneet.agarwal@citrix.com
azure-network-for-cloudbridge-tunnel preview
DASHBOARD
CONFIGURE
Azure-Network-for-
AzureToPratap
AzureVPN
DevTest-Lab
DevTest2
DevTestVirtual
virtual network
Azure-Network-for-
Azure-Subnet-1
GatewaySubnet
GATEWAY
VPN
My-Datacenter-
resources
NAME
ROLE
CLOUD SERVICE
IP ADDRESS
SUBNET NAME
SUBNET
NEW
DOWNLOAD
MANAGE KEY
DISCONNECT
DELETE GATEWAY
EXPORT

DATA IN 3.52KB

DATA OUT 513.65KB

GATEWAY IP ADDRESS 168.63.252.130

Configuring CloudBridge Connector tunnel between datacenter and softlayer enterprise cloud

September 21, 2020

The GUI includes a wizard that helps you to easily configure a CloudBridge Connector tunnel between a NetScaler appliance in a datacenter and NetScaler VPX instances on the SoftLayer enterprise cloud.

When you use the wizard of the NetScaler appliance in the datacenter, the CloudBridge Connector tunnel configuration created on the NetScaler appliance, is automatically pushed to the other endpoint or peer (the NetScaler VPX on SoftLayer) of the CloudBridge Connector tunnel.

Using the wizard of the NetScaler appliance in the datacenter, you perform the following steps to configure a CloudBridge Connector tunnel.

1. Connect to the Softlayer enterprise cloud by providing the user log on credentials.
2. Select the Citrix XenServer that is running the NetScaler VPX appliance.
3. Select the NetScaler VPX appliance.
4. Provide CloudBridge Connector tunnel parameters to:
    - Configure a GRE Tunnel.
    - Configure IPsec on the GRE tunnel.
    - Create a netbridge, which is a logical representation of the CloudBridge connector, by specifying a name.

- Bind the GRE Tunnel to the netbridge.

To configure a CloudBridge Connector tunnel by using the GUI

1. Log on to the GUI of the NetScaler appliance in the datacenter by using your account credentials for the appliance.
2. Navigate to **System** > **CloudBridge Connector** .
3. In the right pane, under **Getting Started**, click **Create/Monitor CloudBridge Connector**.
4. Click **Get Started**.

   > **Note**:
   >
   > If you already have any CloudBridge Connector tunnel configured on the NetScaler appliance, this screen does not appear, and you are taken to the CloudBridge Connector Setup pane.

5. In the CloudBridge Connector Setup pane, click Softlayer, and then follow the instructions in the wizard.

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a NetScaler appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring a CloudBridge Connector tunnel between a NetScaler appliance and Cisco IOS device

September 21, 2020

You can configure a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco device to connect two datacenters or extend your network to a Cloud provider. The NetScaler appliance and the Cisco IOS device form the end points of the CloudBridge Connector tunnel and are called peers.

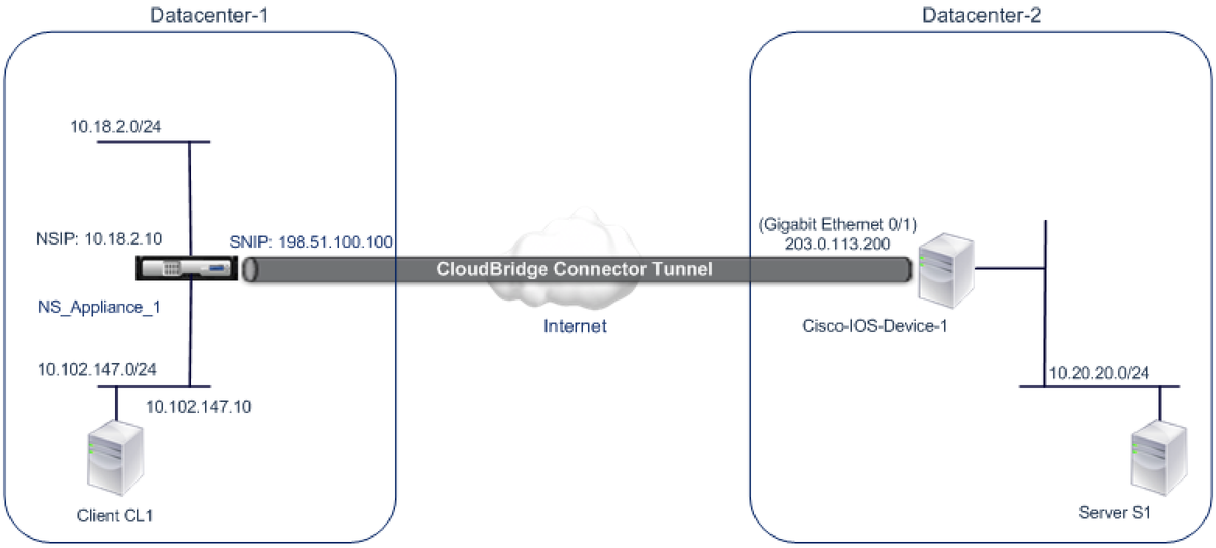
Example of CloudBridge Connector tunnel configuration and data flow

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- NetScaler appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- Cisco IOS device Cisco-IOS-Device-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and Cisco-IOS-Device-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Cisco_IPSec_Profile, CloudBridge Connector tunnel entity NS_Cisco_Tunnel, and policy based routing (PBR) entity NS_Cisco_Pbr.


Datacenter-1
Datacenter-2
10.18.2.0/24
NSIP: 10.18.2.10
SNIP: 198.51.100.100
NS_Appliance_1
10.102.147.0/24
10.102.147.10
Client CL1
CloudBridge Connector Tunnel
Internet
(Gigabit Ethernet 0/1)
203.0.113.200
Cisco-IOS-Device-1
10.20.20.0/24
Server S1

For more information, refer to the CloudBridge Connector tunnel between a NetScaler appliance and Cisco IOS device settings pdf.

Points to Consider for a CloudBridge Connector tunnel configuration

Before configuring a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device, consider the following points:

- The following IPSec settings are supported for a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device.

IPSec Properties	Setting
IPSec mode	Tunnel mode
IKE version	Version 1
IKE authentication method	Pre-Shared Key

IPSec Properties	Setting
IKE encryption algorithm	AES, 3DES
IKE hash algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA384, HMAC SHA512, HMAC MD5
ESP encryption algorithm	AES, 3DES
ESP hash algorithm	HMAC SHA1, HMAC SHA256, HMAC SHA256, HMAC SHA256, HMAC MD5

- You must specify the same IPSec settings on the NetScaler appliance and the Cisco IOS device at the two ends of the CloudBridge Connector.
- NetScaler provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another, and a common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore on the Cisco device, you must specify the same hash algorithm and same encryption algorithm for IKE (while creating IKE policy) and ESP (while creating IPSec transform set).
- You must configure the firewall at the NetScaler end and Cisco device end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets

Configuring Cisco IOS device for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel on a Cisco IOS device, use the Cisco IOS command line interface, which is the primary user interface for configuring, monitoring, and maintaining Cisco devices.

Before you begin the CloudBridge Connector tunnel configuration on a Cisco IOS device, make sure that:

- You have a user account with administrator credentials on the Cisco IOS device.
- You are familiar with the Cisco IOS command line interface.
- The Cisco IOS device is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

> **Note**:
>
> The procedures for configuring CloudBridge Connector tunnel on a Cisco IOS device might change over time, depending on the Cisco release cycle. Citrix recommends that you follow the official Cisco product documention for more information, see Configuring IPSec VPN tunnels

> topic.

To configure a CloudBridge connector tunnel between a NetScaler appliance and a Cisco IOS device, perform the following tasks on the Cisco device's IOS command line:

- **Create an IKE Policy**. An IKE policy defines a combination of security parameters to be used during the IKE negotiation. For example, parameters such as hash algorithm, encryption algorithm, Diffie-Hellman group, and authentication method to be used in the IKE negotiation are set in this task.
- **Configure a Pre-shared key for IKE authentication**. A pre-shared key is a text string, which the peers of a CloudBridge Connector tunnel use to mutually authenticate with each other. The pre-shared keys are matched against each other for IKE authentication. Therefore, for the authentication to be successful, you must configure the same pre-shared key on the Cisco device and the NetScaler appliance.
- **Define a transform set and configure IPSec in tunnel mode**. A transform set defines a combination of security parameters to be used in the exchange of data over the CloudBridge Connector tunnel after the IKE negotiation is successful. Parameters such as hash algorithm and encryption algorithm are set in this task. This task also specifies that IPSec in tunnel mode be used for the CloudBridge tunnel.
- **Create a crypto access List**. Crypto access lists are used to define the subnets whose IP traffic will be protected over the CloudBridge tunnel. The source and destination parameters in the access list specify the Cisco device side and NetScaler side subnets that are to be protected over the CloudBridge Connector Tunnel. The access list must be set to permit. Any request packet that originates from a device in the Cisco device side subnet and is destined to a device in the NetScaler side subnet, and that matches the source and destination parameters of the access list, is sent across the CloudBridge Connector tunnel.
- **Create a crypto map**. Crypto maps define the IPSec parameters to be negotiated with peer. They include the following: Crypto access list to identify the subnets whose traffic is to be protected over the CloudBridge tunnel, peer (NetScaler) identification by IP address, and transform set to match the peer security settings.
- **Apply the crypto Map to an interface**. In this task, you apply a crypto map to an interface through which CloudBridge Connector tunnel traffic will flow. Applying the crypto map to an interface instructs the Cisco IOS device to evaluate all the interface traffic against the crypto map set, and to use the specified policy during connection or SA negotiation on behalf of subnet's traffic to be protected.

The examples in the following procedures create settings of Cisco IOS device Cisco-IOS-Device-1 used in "Example of CloudBridge Connector Configuration and Data Flow."

To create an IKE policy, refer to the IKE policy pdf.

To configure a pre-shared key by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto isakmp identity address	Cisco-ios-device-1(config)# crypto isakmp identity address	Specify the ISAKMP identity (address) for the Cisco IOS device to use when communicating with the peer (NetScaler appliance) during IKE negotiations. This example specifies the address keyword, which uses IP address 203.0.113.200 (Gigabit Ethernet interface 0/1 of Cisco-IOS-Device-1) as the identity for the device.
crypto isakmp key keystringaddress peer-address	Cisco-ios-device-1 (config)# crypto isakmp key examplepresharedkey address 198.51.100.100	Specify a pre-shared key for the IKE authentication. This example configures shared key examplepresharedkey to be used with the NetScaler appliance NS_Appliance-1 (198.51.100.100). The same pre-shared key must be configured on the NetScaler appliance for IKE authentication to be successful between the Cisco IOS device and the NetScaler appliance.

To create a crypto access list by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following command in global configuration mode, in the order shown:

Command	Example	Command Description
access-listaccess-list-number permit IPsource source-wildcard destination destination-wildcard	Cisco-ios-device-1(config)# access-list 111 permit ip 10.20.20.0 0.0.0.255 10.102.14	Specify conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example configures access list 111 to protect traffic from subnets 10.20.20.0/24 (at the Cisco-IOS-Device-1 side) and 10.102.147.0/24 (at the NS_Appliance-1 side).

To define a transform and configure IPSec tunnel mode by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands, starting in global configuration mode, in the order shown:
|Command|Example|Command Description|
|–|–|–|
|crypto ipsec transform-setname ESP_Authentication_Transform ESP_Encryption_Transform
Note: ESP_Authentication_Transform can take the following values:

esp-sha-hmac
esp-sha256-hmac
esp-sha384-hmac
esp-sha512-hmac
esp-md5-hmac

ESP_Encryption_Transformcan take the following values:

esp-aes
esp-3des|Cisco-ios-device-1(config)# crypto ipsec transform-set NS-CISCO-TS esp-sha256-hmac esp-3des|Define a transform set and specify the ESP hash algorithm (for authentication) and the ESP encryption algorithm to be used during exchange of data between the CloudBridge Connector tunnel peers. This example defines transform set NS-CISCO-TS and specifies ESP authentication algorithm as esp-sha256-hmac, and ESP encryption algorithm as esp-3des.|
|mode tunnel|Cisco-ios-device-1 (config-crypto-trans)# mode tunnel|Set IPSec in tunnel mode.|
|exit|Cisco-ios-device-1 (config-crypto-trans)# exit, Cisco-ios-device-1 (config)#|Exit back to global configuration mode.|

To create a crypto map by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
crypto mapmap-name seq-num ipsec-isakmp	Cisco-ios-device-1 (config)# crypto map NS-CISCO-CM 2 ipsec-isakmp	Enter crypto map configuration mode, specify a sequence number for the crypto map, and configure the crypto map to use IKE to establish security associations (SAs). This example configures sequence number 2 and IKE for crypto map NS-CISCO-CM.
set peer ip-address	Cisco-ios-device-1 (config-crypto-map)# set peer 172.23.2.7	Specify the peer (NetScaler appliance) by its IP address. This example specifies 198.51.100.100, which is the CloudBridge Connector endpoint IP address on the NetScaler appliance.
match addressaccess-list-id	Cisco-ios-device-1 (config-crypto-map)# match address 111	Specify an extended access list. This access list specifies conditions to determine the subnets whose IP traffic is to be protected over the CloudBridge Connector tunnel. This example specifies access list 111.
set transform-set transform-set-name	Cisco-ios-device-1 (config-crypto-map)# set transform-set NS-CISCO-TS	Specify which transform sets are allowed for this crypto map entry. This example specifies transform set NS-CISCO-TS.
exit	Cisco-ios-device-1 (config-crypto-map)# exit	

Command	Example	Command Description
Cisco-ios-device-1 (config)#	Exit back to global configuration mode.	

To apply a crypto map to an interface by using the Cisco IOS command line

At the Cisco IOS device's command prompt, type the following commands starting in global configuration mode, in the order shown:

Command	Example	Command Description
interfaceinterface-ID	Cisco-ios-device-1(config)# interface GigabitEthernet 0/1	Specify a physical interface to which to apply the crypto map and enter interface configuration mode. This example specifies Gigabit Ethernet interface 0/1 of the Cisco device Cisco-IOS-Device-1. IP address 203.0.113.200 is already set to this interface.
crypto mapmap-name	Cisco-ios-device-1 (config-if)# crypto map NS-CISCO-CM	Apply the crypto map to the physical interface. This example applies crypto map NS-CISCO-CM.
exit	Cisco-ios-device-1 (config-if)# exit, Cisco-ios-device-1 (config)#	Exit back to global configuration mode.

Configuring the NetScaler appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a NetScaler appliance and a Cisco IOS device, perform the following tasks on the NetScaler appliance. You can use either the NetScaler command line or the NetScaler graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method (Pre-Shared key) to be used by the IPSec protocol in the CloudBridge Connector tunnel.

- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel end point IP address (of type SNIP) configured on the NetScaler appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the Cisco IOS device), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.
- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the NetScaler-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the Cisco IOS device side subnet whose traffic is to be protected over the tunnel. Any request packet that originates from a client in the subnet on the NetScaler side and is destined to a server in the Cisco IOS device side subnet, and matches the source and destination IP range of the PBR entity, is sent across the CloudBridge Connector tunnel associated with the PBR entity. Apply the PBR rule to make it functional.

To create an IPSEC profile by using the NetScaler command line

At the Command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line

At the Command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName <string>`
- `add ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

At the Command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbrs <pbrName>`

The following commands create settings of NetScaler appliance NS_Appliance-1 in "Example of CloudBridge Connector Configuration and Data Flow."

```
> add ipsec profile NS_Cisco_IPSec_Profile -psk examplepresharedkey -ikeVersion v1 - lifetime 315360 - encAlgo 3DES
Done
```

```
    >  add iptunnel NS_Cisco_Tunnel 203.0.113.200 255.255.255.255 198.51.100.100  - protocol IPSEC  - ipsecProfileName NS_Cisco_IPSec_Profile

    Done
    > add pbr NS_Cisco_Pbr -srcIP 10.102.147.0-10.102.147.255  - destIP 10.20.0.0-10.20.255.255  - ipTunnel NS_Cisco_Tunnel

    Done
    > apply pbrs

    Done
<!--NeedCopy-->
```

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** dialog box, set the following parameters:
    - Name
    - Encryption Algorithm
    - Hash Algorithm
    - IKE Protocol Version
4. Configure the **IPSec authentication** method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the **Pre-shared key authentication** method and set the **Pre-Shared Key Exists** parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** dialog box, set the following parameters:
    - Name
    - Remote IP
    - Remote Mask
    - Local IP Type (In the Local IP Type drop down list, select Subnet IP).
    - Local IP (All the configured IPs of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
    - Protocol
    - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** dialog box, set the following parameters:
    - Name
    - Action
    - Next Hop Type (Select IP Tunnel)
    - IP Tunnel Name
    - Source IP Low
    - Source IP High
    - Destination IP Low
    - Destination IP High
4. Click **Create**, and then click **Close**.

To apply a PBR by using the GUI

1. Navigate to **System** > **Network** > **PBRs**.
2. On the **PBRs** tab, select the **PBR**, in the **Action list**, select **Apply**.

The corresponding new CloudBridge Connector tunnel configuration on the NetScaler appliance appears in the GUI. The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

Monitoring the CloudBridge Connector Tunnel

You can monitor the performance of CloudBridge Connector tunnels on a NetScaler appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see Monitoring CloudBridge Connector Tunnels.

Configuring a CloudBridge Connector tunnel between a NetScaler appliance and fortinet fortiGate appliance

September 21, 2020

You can configure a CloudBridge Connector tunnel between a NetScaler appliance and a Fortinet FortiGate appliance to connect two datacenters or extend your network to a cloud provider. The NetScaler appliance and the FortiGate appliance form the end points of the CloudBridge Connector tunnel and are called peers.

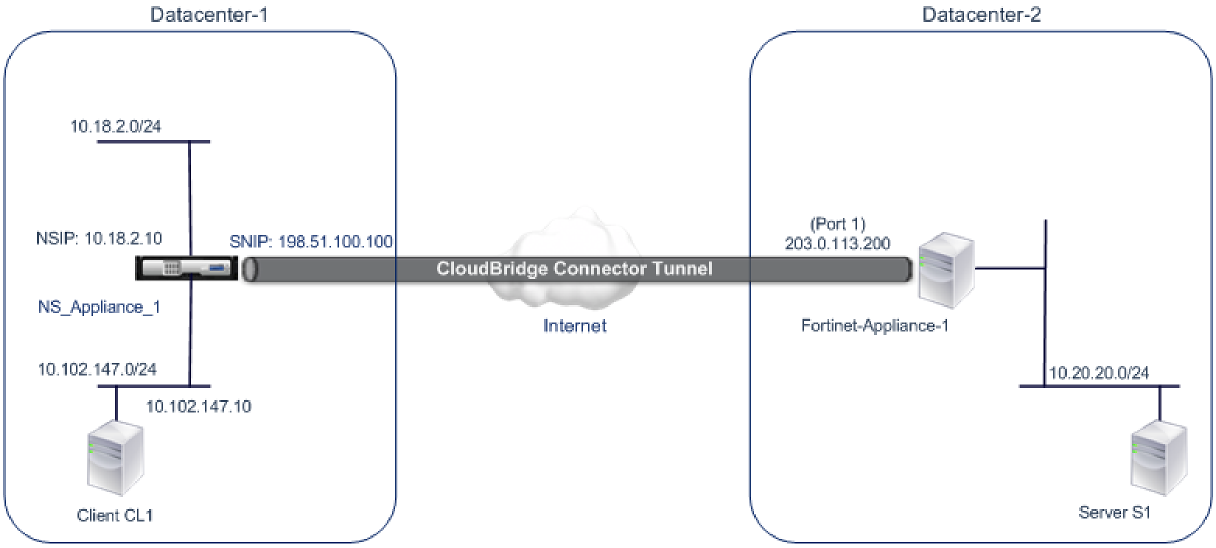
Example of a CloudBridge Connector Tunnel Configuration

As an illustration of the traffic flow in a CloudBridge Connector tunnel, consider an example in which a CloudBridge Connector tunnel is set up between the following devices:

- NetScaler appliance NS_Appliance-1 in a datacenter designated as Datacenter-1
- FortiGate appliance FortiGate-Appliance-1 in a datacenter designated as Datacenter-2

NS_Appliance-1 and FortiGate-Appliance-1 enable communication between private networks in Datacenter-1 and Datacenter-2 through the CloudBridge Connector tunnel. In the example, NS_Appliance-1 and FortiGate-Appliance-1 enable communication between client CL1 in Datacenter-1 and server S1 in Datacenter-2 through the CloudBridge Connector tunnel. Client CL1 and server S1 are on different private networks.

On NS_Appliance-1, the CloudBridge Connector tunnel configuration includes IPSec profile entity NS_Fortinet_IPSec_Profile, CloudBridge Connector tunnel entity NS_Fortinet_Tunnel, and policy based routing (PBR) entity NS_Fortinet_Pbr.


Datacenter-1
Datacenter-2
10.18.2.0/24
NSIP: 10.18.2.10
SNIP: 198.51.100.100
NS_Appliance_1
10.102.147.0/24
10.102.147.10
Client CL1
CloudBridge Connector Tunnel
Internet
(Port 1)
203.0.113.200
Fortinet-Appliance-1
10.20.20.0/24
Server S1

For more information, see CloudBridge Connector tunnel configuration table pdf.

For information about Settings on Fortinet FortiGate-Appliance-1 in Datacenter-2, see table.

Points to consider for a CloudBridge Connector tunnel configuration

The following IPSec settings are supported for a CloudBridge Connector tunnel between a NetScaler appliance and a FortiGate appliance.

IPSec Properties	Setting

IPSec mode	Tunnel mode
IKE version	Version 1
IKE authentication method	Pre-Shared Key
IKE encryption algorithm	AES
IKE hash algorithm	HMAC SHA1
ESP encryption algorithm	AES
ESP hash algorithm	HMAC SHA1

- You must specify the same IPSec settings on the NetScaler appliance and the FortiGate appliance at the two ends of the CloudBridge Connector.
- NetScaler provides a common parameter (in IPSec profiles) for specifying an IKE hash algorithm and an ESP hash algorithm. It also provides another common parameter for specifying an IKE encryption algorithm and an ESP encryption algorithm. Therefore in the FortiGate appliance, you must specify the same hash algorithm and same encryption algorithm in IKE (phase 1 configuration) and ESP (phase 2 configuration).
- You must configure the firewall at the NetScaler end and FortiGate end to allow the following.
  - Any UDP packets for port 500
  - Any UDP packets for port 4500
  - Any ESP (IP protocol number 50) packets
- FortiGate appliance supports two types of VPN tunnels: Policy-based and Route-based. Only policy-based VPN tunnel is supported between a FortiGate appliance and a NetScaler appliance.

Configuring FortiGate appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel on a FortiGate appliance, use the Fortinet Web-based Manager, which is the primary user interface for configuring, monitoring, and maintaining FortiGate appliances.

Before you begin the CloudBridge Connector tunnel configuration on a FortiGate appliance, make sure that:

- You have a user account with administrator credentials on the FortiGate appliance.
- You are familiar with the Fortinet Web-based Manager.
- The FortiGate appliance is UP and running, is connected to the Internet, and is also connected to the private subnets whose traffic is to be protected over the CloudBridge Connector tunnel.

> **Note**
>
> The procedures for configuring CloudBridge Connector tunnel on a FortiGate appliance might change over time, depending on the Fortinet release cycle. Citrix recommends that you follow the official Fortinet product documentation for Configuring IPSec VPN tunnels.

To configure a CloudBridge connector tunnel between a NetScaler appliance and a FortiGate appliance, perform the following tasks on the FortiGate appliance by using the Fortinet Web-based manager:

- **Enable Policy-based IPSec VPN feature**. Enable this feature for creating policy-based VPN tunnels on the FortiGate appliance. Only policy-based type of VPN tunnel is supported between a FortiGate appliance and a NetScaler appliance. A policy-based VPN tunnel configuration on a FortiGate appliance includes phase 1 settings, phase 2 settings, and an IPSec security policy.
- **Define phase 1 parameters.** Phase 1 parameters are used by the FortiGate appliance for IKE Authentication before forming a secure tunnel to the NetScaler appliance.
- **Define phase 2 parameters**. Phase 2 parameters are used by the FortiGate appliance for forming a secure tunnel to the NetScaler appliance by establishing IKE security associations (SA).
- **Specify private subnets**. Define the FortiGate-side and the NetScaler-side private subnets whose IP traffic is to be transported through the tunnel.
- **Define an IPSec security policy for the tunnel**. A security policy allow IP traffic to pass between interfaces on a FortiGate appliance. An IPSec security policy specifies the interface to the private subnet and the interface connecting the NetScaler appliance through the tunnel.

To enable Policy-based IPSec VPN feature by using the Fortinet Web-based Manager

1. Navigate to **System** > **Config** > **Features**.
2. On the **Feature Settings** page, select **Show More** and turn on **Policy-based IPSec VPN**.

To define phase 1 parameters by using the Fortinet Web-based Manager

1. Navigate to **VPN** > **IPsec** > **Auto Key (IKE)** and click **Create Phase1**.
2. On the **New Phase 1** page, set the following parameters:
    - Name: Enter a name for this phase 1 configuration.
    - Remote Gateway: Select *Static IP Address.*
    - Mode: Select *Main (ID Protection)*.
    - Authentication Method: Select *Preshared Key*.
    - Pre-Shared Key: Enter a pre-shared key. The same pre-shared key must be configured on the NetScaler appliance.
    - Peer Options: Set the following IKE paramters for authenticating a NetScaler appliance.
        - IKE Version: Select *1*.
        - Mode Config: Clear this option if it is selected.
        - Local Gateway IP: Select *Main Interface IP*.

        - P1 Proposal: Select the encryption and authentication algorithms for IKE Authentication before forming a secure tunnel to the NetScaler appliance.
            * 1 - Encryption: Select *AES128*.
            * Authentication: Select *SHA1*.
            * Keylife: Enter an amount of time (in seconds) for the phase 1 key life.
            * DH Group: Select 2.
        - X-Auth: Select *Disable*.
        - Deed Peer Detection: Select this option.
3. Click **OK**.

To specify private subnets by using the Fortinet Web-based Manager

1. Navigate to **Firewall Objects** > **Address** > **Addresses** and select **Create New**.
2. On the **New Address** page, set the following parameters:
    - Name: Enter a name for FortiGate-side subnet.
    - Type: Select *Subnet*.
    - Subnet / IP Range: Enter the address of the FortiGate-side subnet.
    - Interface: Select the local interface to this subnet.
3. Click **OK**.
4. Repeat steps 1-3 to specify the NetScaler-side subnet.

To define phase 2 parameters by using the Fortinet Web-based Manager

1. Navigate to **VPN** > **IPsec** > **Auto Key (IKE)** and click **Create Phase 2**.
2. On the **New Phase 2** page, set the following parameters:
    - Name: Enter a name for this phase 2 configuration.
    - Phase 1: Select the Phase 1 configuration from the drop-down list.
3. Click **Advanced** and set the following parameters:
    - P2 Proposal: Select the encryption and authentication algorithms for forming a secure tunnel to the NetScaler appliance.
        - 1 - Encryption: Select *AES128*.
        - Authentication: Select *SHA1*.
        - Enable replay detection: Select this option.
        - Enable perfect forward secrecy (PFS): Select this option.
        - DH Group: Select *2*.
    - Keylife: Enter an amount of time (in seconds) for the phase 2 key life.
    - Autokey Keep Alive: Select this option.
    - Auto-negotiate: Select this option.
    - Quick Mode Selector: Specify the FortiGate-side and the NetScaler-side private subnets whose traffic is to be traversed through the tunnel.
        - Source Address: Select the FortiGate-side subnet from the drop-down list.
        - Source Port: Enter *0*.

    - Destination Address: Select the NetScaler-side subnet from the drop-down list.
    - Destination Port: Enter *0*.
    - Protocol: Enter *0*.
4. Click **OK**.

To define an IPSec security policy by using the Fortinet Web-based Manager

1. Navigate to **Policy** > **Policy** > **Policy** and click **Create New**.
2. On the **Edit Policy** page, set the following parameters:
    - Policy Type: Select *VPN*.
    - Policy Subtype: Select *IPSec*.
    - Local Interface: Select the local interface to the internal (private) network.
    - Local Protected Subnet: Select the FortiGate-side subnet from the drop-down list whose traffic is to be traversed through the tunnel.
    - Outgoing VPN Interface: Select the local interface to the external (public) network.
    - Remote Protected Subnet: Select the NetScaler-side subnet from the drop-down list whose traffic is to be traversed through the tunnel.
    - Schedule: Keep the default setting (*always*) unless changes are needed to meet specific requirements.
    - Service: Keep the default setting (*ANY*) unless changes are needed to meet your specific requirements.
    - VPN Tunnel: Select *Use Existing* and select the tunnel from the drop-down list.
    - Allow traffic to be initiated from the remote site: Select if traffic from the remote network will be allowed to initiate the tunnel.
3. Click **OK**.

Configuring the NetScaler appliance for the CloudBridge Connector tunnel

To configure a CloudBridge Connector tunnel between a NetScaler appliance and a FortiGate appliance, perform the following tasks on the NetScaler appliance. You can use either the NetScaler command line or the NetScaler graphical user interface (GUI):

- **Create an IPSec profile**. An IPSec profile entity specifies the IPSec protocol parameters, such as IKE version, encryption algorithm, hash algorithm, and authentication method to be used by the IPSec protocol in the CloudBridge Connector tunnel.
- **Create an IP tunnel that uses IPSec protocol, and associate the IPSec profile with it**. An IP tunnel specifies the local IP address (CloudBridge Connector tunnel end point IP address (of type SNIP) configured on the NetScaler appliance), remote IP address (CloudBridge Connector tunnel endpoint IP address configured on the FortiGate appliance), protocol (IPSec) used to set up the CloudBridge Connector tunnel, and an IPSec profile entity. The created IP tunnel entity is also called the CloudBridge Connector tunnel entity.

- **Create a PBR rule and associate it with the IP tunnel**. A PBR entity specifies a set of rules and an IP tunnel (CloudBridge Connector tunnel) entity. The source IP address range and the destination IP address range are the conditions for the PBR entity. Set the source IP address range to specify the NetScaler-side subnet whose traffic is to be protected over the tunnel, and set the destination IP address range to specify the FortiGate appliance side subnet whose traffic is to be protected over the tunnel.

To create an IPSEC profile by using the NetScaler command line

At the command prompt, type:

- `add ipsec profile <name> -psk <string> -ikeVersion v1 -encAlgo AES -hashAlgo HMAC_SHA1 -perfectForwardSecrecy ENABLE`
- `show ipsec profile <name>`

To create an IPSEC tunnel and bind the IPSEC profile to it by using the NetScaler command line

At the command prompt, type:

- `add ipTunnel <name> <remote> <remoteSubnetMask> <local> -protocol IPSEC -ipsecProfileName** <string>`
- `show ipTunnel <name>`

To create a PBR rule and bind the IPSEC tunnel to it by using the NetScaler command line

At the command prompt, type:

- `add pbr <pbrName> ALLOW -srcIP <subnet-range> -destIP <subnet-range> -ipTunnel <tunnelName>`
- `apply pbrs`
- `show pbr <pbrName>`

To create an IPSEC profile by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IPSec Profile**.
2. In the details pane, click **Add**.
3. In the **Add IPSec Profile** page, set the following parameters:
    - Name
    - Encryption Algorithm
    - Hash Algorithm
    - IKE Protocol Version
    - Perfect Forward Secrecy (Enable this paramter)
4. Configure the IPSec authentication method to be used by the two CloudBridge Connector tunnel peers to mutually authenticate: Select the Pre-shared key authentication method and set the Pre-Shared Key Exists parameter.
5. Click **Create**, and then click **Close**.

To create an IP tunnel and bind the IPSEC profile to it by using the GUI

1. Navigate to **System** > **CloudBridge Connector** > **IP Tunnels**.
2. On the **IPv4 Tunnels** tab, click **Add**.
3. In the **Add IP Tunnel** page, set the following parameters:
    - Name
    - Remote IP
    - Remote Mask
    - Local IP Type (In the Local IP Type drop-down list, select *Subnet IP*).
    - Local IP (All the configured IP addresses of the selected IP type are in the Local IP drop down list. Select the desired IP from the list.)
    - Protocol
    - IPSec Profile
4. Click **Create**, and then click **Close**.

To create a PBR rule and bind the IPSEC tunnel to it by using the GUI

1. Navigate to **System** > **Network** > **PBR**.
2. On the **PBR** tab, click **Add**.
3. In the **Create PBR** page, set the following parameters:
    - Name
    - Action
    - Next Hop Type (Select *IP Tunnel*)
    - IP Tunnel Name
    - Source IP Low
    - Source IP High
    - Destination IP Low
    - Destination IP High
4. Click **Create**, and then click **Close**.

The corresponding new CloudBridge Connector tunnel configuration on the NetScaler appliance appears in the GUI.

The current status of the CloudBridge connector tunnel is shown in the Configured CloudBridge Connector pane. A green dot indicates that the tunnel is up. A red dot indicates that the tunnel is down.

The following commands create settings of NetScaler appliance NS_Appliance-1 in "Example of a CloudBridge Connector Configuration."

```
    >  add ipsec profile NS_Fortinet_IPSec_Profile -psk examplepresharedkey -ikeVersion v1  - encAlgo AES  - hashalgo HMAC_SHA1  - lifetime 315360 -perfectForwardSecrecy ENABLE

     Done
    >  add iptunnel NS_Fortinet_Tunnel 203.0.113.200 255.255.255.255 198.51.100.100 -protocol IPSEC -ipsecProfileName NS_Fortinet_IPSec_Profile

     Done
    > add pbr NS_Fortinet_Pbr -srcIP 10.102.147.0-10.102.147.255 -destIP 10.20.0.0-10.20.255.255 -ipTunnel NS_Fortinet_Tunnel

     Done
    > apply pbrs

     Done
<!--NeedCopy-->
```

Monitoring the CloudBridge Connector tunnel

You can monitor the performance of CloudBridge Connector tunnels on a NetScaler appliance by using CloudBridge Connector tunnel statistical counters. For more information about displaying CloudBridge Connector tunnel statistics on a NetScaler appliance, see Monitoring CloudBridge Connector Tunnels.

CloudBridge Connector tunnel diagnostics and troubleshooting

September 21, 2020

If you have problems with a CloudBridge Connector tunnel configuration, make sure that all prerequisites were observed before the tunnel was set up. If they were, the problem might be with the tunnel end-point IP addresses, a NAT configuration, the way the tunnel was set up, or with the data traffic.

Troubleshooting a CloudBridge Connector tunnel

If your CloudBridge Connector tunnel does not function properly, the issue could be with tunnel establishment or with the data traffic. If you are unsure which type of problem you have, look for an error message in the log file and see if the error message is in the list of tunnel-establishment issues. If you do not find your error message, check the list of possible issues related to data traffic.

Issues Related to tunnel establishment

After the requirements for configuring the IPSec tunnel are met and the CloudBridge Connector tunnel is configured, if the status of the tunnel is not UP, look for debugging information in the iked.log file on one or both NetScaler appliances configured as the tunnel end points.

On either appliance, type the following command at the NetScaler shell prompt:

```
'cat /tmp/iked.debug                    tee /var/iked.log'
```

The Troubleshooting pdf lists some common errors and their solutions.

Issues related to data traffic

If the data in the CloudBridge Connector tunnel are not exchanged properly between the tunnel end points, do the following.

- For a CloudBridge Connector tunnel that uses GRE and IPSec protocols:
  - Make sure that L2 mode is enabled on both of the CloudBridge Connector tunnel end points. To enable L2 mode, type the following command at the NetScaler command line interface:

    `enable mode L2`

    - If one of the CloudBridge Connector tunnel end points is a CloudBridge virtual appliance (VPX) and is provisioned on a VMware ESXi hypervisor, make sure that Promiscuous mode is set to Accept for the vSwitch associated with the CloudBridge VPX appliance.
  - If a VLAN is extended through a CloudBridge Connector tunnel, verify one-to-one mapping on the extended VLAN entity on each of the tunnel end points
  - Make sure that the IP tunnel entity is bound to the correct netbridge entity in each of the tunnel end points.
  - Verify that the ARP entry for the peer CloudBridge Connector tunnel end point exists on the local tunnel end point, by typing the following command at the NetScaler command line interface:

    'show arp'
  - If the output shows an incomplete ARP entry, bidirectional traffic is not flowing through the tunnel. If bidirectional traffic is flowing, the ARP entry shows the name of tunnel interface for the devices on the other side of the tunnel.
  - Remove the IP tunnel entities from both tunnel end points and add them again with the same parameters, but with the IPSec profile set to NONE, so that the tunnel uses only the GRE protocol.

    After verifying the following in the IP tunnel (that uses GRE protocol), configure the tunnel with IPSec parameters by specifying a valid IPSec profile to the respective IP tunnel entities on each of the tunnel end points.

    Proper PING or TCP flow through the tunnel.

Proper flow of data traffic through the tunnel.
After the configured tunnel (that uses GRE and IPSec protocols) is in UP state, if the data traffic does not flow properly through the tunnel, and if a NAT device was deployed in front of any or both of the tunnel end points, analyze the ingress and egress packets on the NAT devices.

- If a NetScaler appliance is used as Router or Gateway.
  - Make sure that L3 mode is enabled on the NetScaler appliance. To enable L3 mode, run the following command in the CloudBridge command line.
  - enable mode L3
  - If subnets are bound to a netbridge entity, make sure that correct IP tunnel entity is also bound to the netbridge.
  - Run the following command in the NetScaler command line to see where the packets (Input and Output)are getting dropped:
    `stat ipsec counters`
  - Make sure that the correct routes are configured on both the tunnel end points.
  - If no NAT device is deployed in front of the NetScaler appliance, make sure that the firewalls are configured to allow any ESP (IP protocol number 50) packets and any UDP packets for port 4500.

If none of the above measures result in successful exchange of traffic between the tunnel end points, contact Citrix Technical Support.

Checklist before contacting Citrix technical Support

For a speedy resolution, make sure that you have the following items ready before contacting Citrix Technical Support.

- Details of the deployment and network topology.
- Log file collected by typing the following command at the NetScaler shell prompt.
  ```
  cat /tmp/iked.debug | tee /var/log/iked.log
  ```
- Tech support bundle captured by typing the following command at the NetScaler command line.
  ```
  show techsupport
  ```
- Packet traces captured on both CloudBridge Connector tunnel end points. To start a packet trace, type the following command at the NetScaler command line.
  ```
  start nstrace -size 0
  ```

  To stop packet trace, type the following command at the NetScaler command line.
  ```
  stop nstrace
  ```

- Output of the following command typed at the NetScaler command prompt.
  ```
  show arp
  ```

High Availability

September 21, 2020

A high availability (HA) deployment of two NetScaler appliances can provide uninterrupted operation in any transaction. With one appliance configured as the primary node and the other as the secondary node, the primary node accepts connections and manages servers while the secondary node monitors the primary. If, for any reason, the primary node is unable to accept connections, the secondary node takes over.

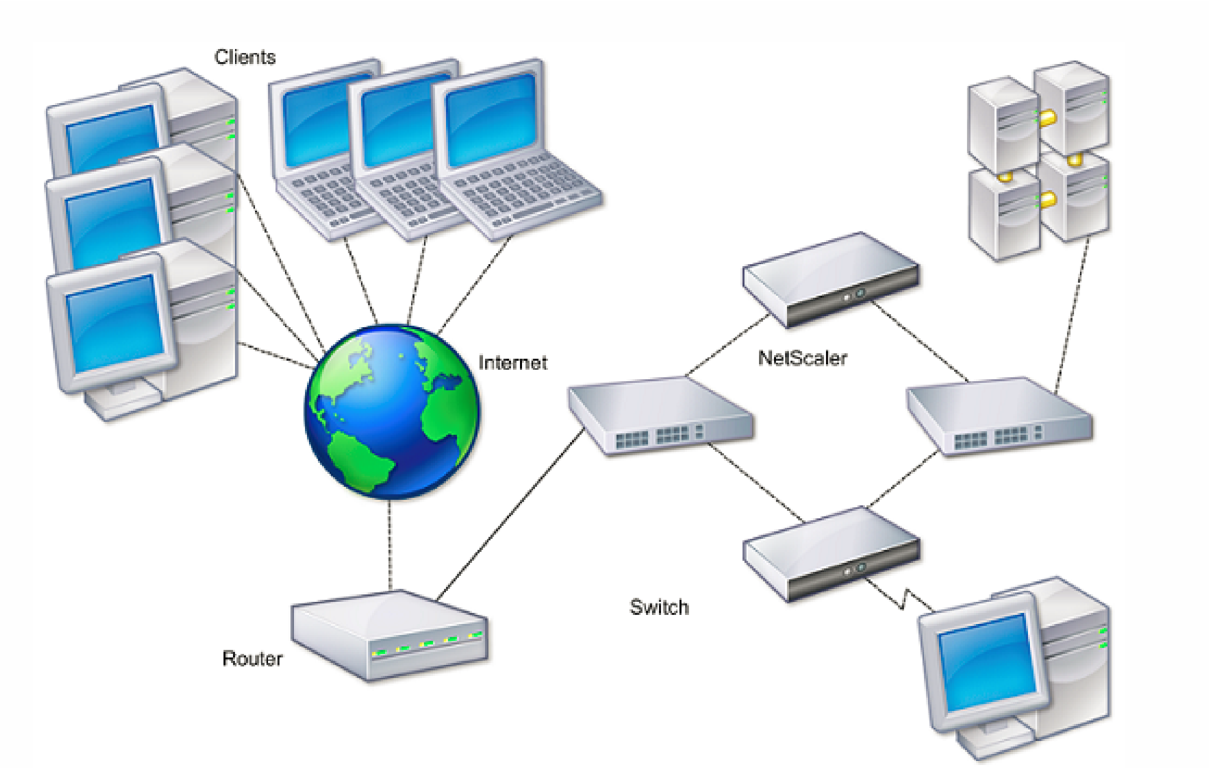
The secondary node monitors the primary by sending periodic messages (often called heartbeat messages or health checks) to determine whether the primary node is accepting connections. If a health check fails, the secondary node retries the connection for a specified period, after which it determines that the primary node is not functioning normally. The secondary node then takes over for the primary (a process called failover).

After a failover, all clients must reestablish their connections to the managed servers, but the session persistence rules are maintained as they were before the failover.

With Web server logging persistence enabled, no log data is lost due to the failover. For logging persistence to be enabled, the log server configuration must carry entries for both systems in the log.conf file.

The following figure shows a network configuration with an HA pair.

Figure 1. NetScaler Appliances in a High Availability Configuration



To configure HA, you might want to begin by creating a basic setup, with both nodes in the same subnet. You can then customize the intervals at which the nodes communicate health-check information, the process by which nodes maintain synchronization, and the propagation of commands from the primary to the secondary. You can configure fail-safe mode to prevent a situation in which neither node is primary. If your environment includes devices that do not accept NetScaler gratuitous ARP messages, you should configure virtual MAC addresses. When you are ready for a more complex configuration, you can configure HA nodes in different subnets.

To improve the reliability of your HA setup, you can configure route monitors and create redundant links. In some situations, such as when troubleshooting or performing maintenance tasks, you might want to force a node to fail over (assign primary status to the other node), or you might want to force the secondary node to stay secondary or the primary node to stay primary.

Points to consider for a high availability setup

September 21, 2020

> **Note**
>
> The following requirements for configuring systems in an HA setup:

- In an HA configuration, the primary and secondary NetScaler appliances should be of the same model. Different NetScaler models are not supported in an HA pair.
- In an HA setup, both nodes must run the same version of NetScaler, for example, nCore/nCore or classic/classic. If the nodes are running NetScaler classic and you want to migrate to NetScaler nCore of the same NetScaler release, prop and sync are not supported during the migration process. Once migration is complete, prop and sync are auto-enabled. The same applies if you migrate from NetScaler nCore to NetScaler classic.
- Entries in the configuration file (ns.conf) on both the primary and the secondary system must match, with the following exceptions:
  - The primary and the secondary systems must each be configured with their own unique NetScaler IP addresses (NSIPs.)
  - In an HA pair, the node ID and associated IP address of one node must point to the other node. For example, if you have nodes NS1 and NS2, you must configure NS1 with a unique node ID and the IP address of NS2, and you must configure NS2 with a unique node ID and the IP address of NS1.
- If you create a configuration file on either node by using a method that does not go directly through the GUI or the CLI (for example, importing SSL certificates, or changing to startup scripts), you must copy the configuration file to the other node or create an identical file on that node.
- Initially, all NetScaler appliances are configured with the same RPC node password. RPC nodes are internal system entities used for system-to-system communication of configuration and session information. For security, you should change the default RPC node passwords.

  One RPC node exists on each NetScaler. This node stores the password, which is checked against the password provided by the contacting system. To communicate with other systems, each NetScaler requires knowledge of those systems, including how to authenticate on those systems. RPC nodes maintain this information, which includes the IP addresses of the other systems, and the passwords they require for authentication.

  RPC nodes are implicitly created when adding a node or adding a Global Server Load Balancing (GSLB) site. You cannot create or delete RPC nodes manually.

  > **Note**:
  >
  > If the NetScaler appliances in a high availability setup are configured in one-arm mode, you must disable all system interfaces except the one connected to the switch or hub.

  For an IPv6 HA configuration, the following considerations apply:

  - You must install the IPv6PT license on both NetScaler appliances.
  - After installing the IPv6PT license, enable the IPv6 feature by using the GUI or the command line interface.

  - Both NetScaler appliances require a global NSIP IPv6 address. In addition, network entities (for example, switches and routers) between the two nodes must support IPv6.

Configuring high availability

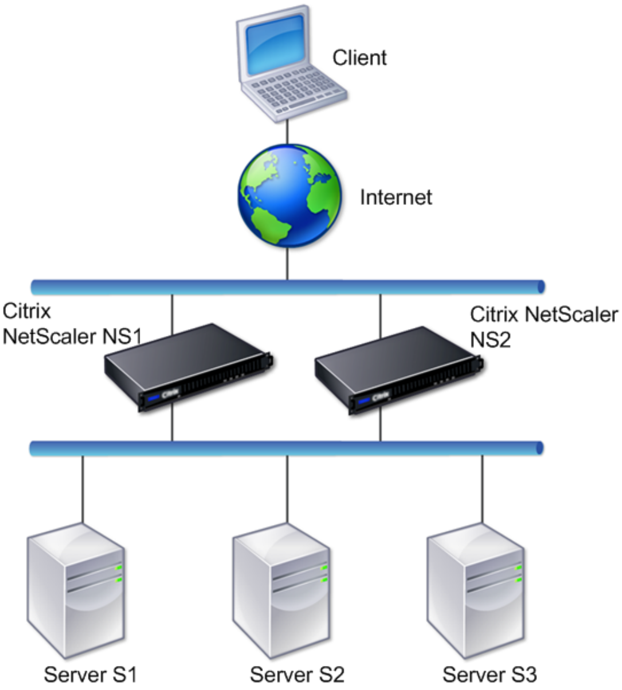
September 21, 2020

To set up a high availability configuration, you create two nodes, each of which defines the other's NetScaler IP (NSIP) address as a remote node. Begin by logging on to one of the two NetScaler appliances that you want to configure for high availability, and add a node. Specify the other appliance's NetScaler IP (NSIP) address as the address of the new node. Then, log on to the other appliance and add a node that has the NSIP address of the first appliance. An algorithm determines which node becomes primary and which becomes secondary.

> **Note**:
>
> The GUI provides an option that avoids having to log on to the second appliance.

The following figure shows a simple HA setup, in which both nodes are in same subnet.

Figure 1. Two NetScaler Appliances Connected in a High Availability Configuration


Client
Internet
Citrix
NetScaler NS1
Citrix NetScaler
NS2
Server S1
Server S2
Server S3

Adding a remote node

To add a remote NetScaler appliance as a node in a high availability setup, you specify a unique node ID and the appliance's NSIP. The maximum number of node IDs in an HA setup is 64. When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

> **Note**:
>
> To ensure that each node in the high availability configuration has the same settings, you should synchronize your SSL certificates, startup scripts, and other configuration files with those on the primary node.

To add a node by using the command line interface

At the command prompt, type:

- `add ha node <id> <IPAddress>`
- `show ha node`

Example

```
> add ha node 10 203.0.113.32
<!--NeedCopy-->
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

Example

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

To add a remote node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, add a new remote node, or edit an existing node.

Disabling or Enabling a Node

You can disable or enable only a secondary node. When you disable a secondary node, it stops sending heartbeat messages to the primary node, and therefore the primary node can no longer check the status of the secondary. When you enable a node, the node takes part in the high availability configuration.

To disable or enable a node by using the command line interface

At the command prompt, type one of the following commands:

- `set ha node -hastatus DISABLED`
- `set ha node -hastatus ENABLED`

To disable or enable a node by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. In the **High Availability Status** list, select **ENABLED (Actively Participate in HA)** or **DISABLED (Do not participate in HA)**.

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

`rm ha node <id>`

Example

```
> rm ha node 10
  Done
<!--NeedCopy-->
```

To remove a node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, delete the node.

Configuring the communication intervals

September 21, 2020

The hello interval is the interval at which the heartbeat messages are sent to the peer node. The dead interval is the time interval after which the peer node is marked DOWN if heartbeat packets are not received. The heartbeat messages are UDP packets sent to port 3003 of the other node in an HA pair. Dead interval must be set as a multiple of hello interval.

To set the hello and dead intervals by using the command line interface

At the command prompt, type:

- `set HA node [-helloInterval <msecs>] [-deadInterval <secs>]`
- `show HA node <id>`

To set the hello and dead intervals by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Set the following parameters:
    - Hello Interval (msecs)
    - Dead Interval (secs)

Configuring synchronization

September 21, 2020

Synchronization is a process of duplicating the configuration of the primary node on the secondary node. The purpose of synchronization is to ensure that there is no loss of configuration information between the primary and the secondary nodes, regardless of the number of failovers that occur. Synchronization uses port 3010.

Synchronization is triggered by either of the following circumstances:

- The secondary node in an HA setup comes up after a restart.
- The primary node becomes secondary after a failover.

Automatic synchronization is enabled by default. You can also force synchronization.

Disabling or enabling synchronization

Automatic HA synchronization is enabled by default on each node in an HA pair. You can enable or disable it on either node.

To disable or enable automatic synchronization by using the command line interface

At the command prompt, type:

- `set HA node -haSync DISABLED`
- `set HA node -haSync ENABLED`

To disable or enable synchronization by using the GUI

1. Navigate to **System** > **High Availability**.
2. Under HA Synchronization, clear or select the Secondary node will fetch the configuration from Primary option.

Forcing the secondary node to synchronize with the primary node

In addition to automatic synchronization, the NetScaler supports forced synchronization. You can force the synchronization from either the primary or the secondary node. When you force synchronization from the secondary node, it starts synchronizing its configuration with the primary node.

However, if synchronization is already in progress, forced synchronization fails and the system displays a warning. Forced synchronization also fails in any of the following circumstances:

- You force synchronization on a standalone system.

- The secondary node is disabled.
- HA synchronization is disabled on the secondary node.

To force synchronization by using the command line interface

At the command prompt, type:

`force HA sync`

To force synchronization by using the GUI

1. Navigate to System > High Availability.
2. On the **Nodes** tab, in the Action list, click **Force Synchronization**.

Synchronizing configuration files in a high availability setup

September 21, 2020

In a high availability setup, all configuration files are synchronised automatically from the primary node to the secondary node at an interval of one minute. Synchronizing configuration files can be performed manually by using the command line interface or the GUI at either the primary or the secondary node.

Files located on the secondary that are specific to the secondary (not present on the primary) are not deleted during the synchronization.

To synchronize files in a high availability setup by using the command line interface

At the command prompt, type:

`sync HA files <mode>`

Example

```
> sync HA files all
 Done
<!--NeedCopy-->
```

```
> sync HA files ssl
 Done
<!--NeedCopy-->
```

Parameter descriptions (of the command listed in the CLI procedure)

`sync ha files <mode>`

`mode`

Specify one of the following modes of synchronization.

- **all** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, and Application Firewall XML objects.
- **bookmarks** - Synchronize all Access Gateway bookmarks.
- **ssl** - Synchronize all certificates, keys, and CRLs for the SSL feature.
- **htmlinjection** - Synchronize all scripts configured for the HTML injection feature.
- **imports** - Synchronizeall XML objects (for example, WSDLs, schemas, error pages) configured for the application firewall.
- **misc** - Synchronize all license files and the rc.conf file.
- **all_plus_misc** - Synchronize files related to system configuration, Access Gateway bookmarks, SSL certificates, SSL CRL lists, HTML injection scripts, application firewall XML objects, licenses, and the rc.conf file.

To synchronize files in a high availability setup by using the GUI

Navigate to **System** > **Diagnostics** and, in the **Utilities** group, click **Start HA files synchronization**.

Configuring command propagation

September 21, 2020

In an HA setup, any command issued on the primary node propagates automatically to, and is executed on, the secondary before it is executed on the primary. If command propagation fails, or if command execution fails on the secondary, the primary node executes the command and logs an error. Command propagation uses port 3010.

In an HA pair configuration, command propagation is enabled by default on both the primary and secondary nodes. You can enable or disable command propagation on either node in an HA pair. If you disable command propagation on the primary node, commands are not propagated to the secondary node. If you disable command propagation on the secondary node, commands propagated from the primary are not executed on the secondary node.

> **Note**
>
> After reenabling propagation, remember to force synchronization.

If synchronization occurs while you are disabling propagation, any configuration-related changes that you make before the disabling of propagation takes effect are synchronized with the secondary node. This is also true for cases where propagation is disabled while synchronization is in progress.

To disable or enable command propagation by using the command line interface

At the command prompt, type:

- set HA node -haProp DISABLED
- set HA node -haProp ENABLED

To disable or enable command propagation by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Clear or select the Primary node will propagate configuration to the Secondary option.

Restricting high availability synchronization traffic to a VLAN

June 17, 2021

In a high availability (HA) deployment, traffic related to maintaining the HA configuration flows between the two HA nodes. This traffic is of the following types:

- Config synchronization
- Config propagation
- Connection mirroring
- Load balancing persistency config synchronization
- Persistent session synchronization
- Session state synchronization

Proper flow of this HA related traffic between the two nodes is critical for the functioning of the HA deployment. Typically, the HA related traffic is small in volume but can become very high during a failover. It becomes very high if stateful connection failover is enabled and the node that was primary before the failover was handling a large number of connections.

By default, the HA related traffic flows through the VLANs to which the NSIP address is bound. To accommodate a potential surge in this traffic, you can separate the HA related traffic from the management traffic and restrict its flow to a separate VLAN. This VLAN is called the HA SYNC VLAN.

Points to consider before Configuring an HA SYNC VLAN

- The configuration of an HA SYNC VLAN is neither propagated nor synchronized. In other words, the HA SYNC VLAN is node specific and is configured independently on each node.
- HA SYNC VLAN configuration is removed when you clear the configuration in only FULL mode.
- HA MON must be set to OFF for interfaces that are part of the HA SYNC VLAN, to avoid a situation in which both nodes function as the primary node.
- Management interfaces (for example, 0/1 and 0/2) must not be part of the HA SYNC VLAN, so that HA related traffic does not flow through management interfaces.
- Citrix recommends disabling high availability heartbeat messages on management interfaces and enabling on HA SYNC VLAN interfaces. After meeting these recommendations, high availability heartbeat messages can also be enabled on data interfaces.

  For more information on disabling high availability heartbeat messages on interfaces, see Managing high availability heartbeat messages on a NetScaler appliance.

To configure an HA SYNC VLAN on a NetScaler node, specify a configured VLAN with the HA SYNC VLAN parameter of the local node entity.

To configure an HA SYNC VLAN on a local node by using the command line:

At the command prompt, type:

- `set ha node -syncvlan <VLANID>`
- `show node`

Parameter Description:

syncvlan (Sync VLAN) - VLAN on which HA related traffic is sent. This includes traffic for synchronization, propagation, connection mirroring, load balancing persistency, configuration synchronization, persistent session synchronization, and session state synchronization. However, HA heartbeats can use any interface.

To configure an HA SYNC VLAN on a node by using the GUI:

1. Navigate to **System** > **High Availability**.
2. Set the **Sync VLAN** parameter while modifying the local node.

Configuring fail-safe mode

September 21, 2020

In an HA configuration, fail-safe mode ensures that one node is always primary when both nodes fail the health check. This is to ensure that when a node is only partially available, backup methods are enabled to handle traffic as best as possible. The HA fail-safe mode is configured independently on each node.

The following table shows some of the fail-safe cases. The NOT_UP state means that the node failed the health check yet it is partially available. The UP state means that the node passed the health check.

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP(failed last)	NOT_UP (failed first)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
NOT_UP (failed first)	NOT_UP(failed last)	A (Secondary), B (Secondary)	A(Secondary), B(Primary)	If both nodes fail, one after the other, the node that was the last primary remains primary.
UP	UP	A (Primary), B (Secondary)	A (Primary), B (Secondary)	If both nodes pass the health check, no change in behavior with fail-safe enabled.
UP	NOT_UP	A(Primary), B(Secondary)	A (Primary), B (Secondary)	If only the secondary node fails, no change in behavior with fail-safe enabled.

Node A (Primary) Health State	Node B (Secondary) Health State	Default HA Behavior	Fail-Safe Enabled HA Behavior	Description
NOT_UP	UP	A(Secondary), B(Primary)	A(Secondary), B(Primary)	If only the primary fails, no change in behavior with fail-safe enabled.
NOT_UP	UP (STAYSECONDARY)	A (Secondary), B (Secondary)	A (Primary), B (Secondary)	If the secondary is configured as STAYSECONDARY, the primary remains primary even if it fails.

To enable fail-safe mode by using the command line interface

At the command prompt, type:

```
set HA node [-failSafe ( ON | OFF )]
```

Example

```
set ha node -failsafe ON
<!--NeedCopy-->
```

To enable fail-safe mode by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the node.
2. Under **Fail-Safe Mode**, select the **Maintain one Primary** node even when both nodes are unhealthy option.

Configuring Virtual MAC Addresses

September 21, 2020

A Virtual MAC address is a floating entity shared by the primary and the secondary nodes in an HA setup.

In an HA setup, the primary node owns all of the floating IP addresses, such as the MIPs, SNIPs, and VIPs. The primary node responds to Address Resolution Protocol (ARP) requests for these IP addresses with its own MAC address. As a result, the ARP table of an external device (for example, an upstream router) is updated with the floating IP address and the primary node's MAC address.

When a failover occurs, the secondary node takes over as the new primary node. It then uses Gratuitous ARP (GARP) to advertise the floating IP addresses that it acquired from the primary. However, the MAC address that the new primary advertises is the MAC address of its own interface.

Some devices (notably a few routers) do not accept the GARP messages generated by the NetScaler appliance. As a result, some external devices retain the old IP to MAC mapping advertised by the old primary node. This can result in a site going down.

You can overcome this problem by configuring a virtual MAC on both nodes of an HA pair. Both nodes then possess identical MAC addresses. Therefore, when failover occurs, the MAC address of the secondary node remains unchanged, and the ARP tables on the external devices do not need to be updated.

To create a Virtual MAC, you need to first create a Virtual Router ID (VRID) and bind it to an interface. (In an HA setup, you need to bind the VRID to the interfaces on both nodes.) Once the VRID is bound to an interface, the system generates a virtual MAC with the VRID as the last octet.

This section includes the following details:

- Configuring IPv4 Virtual MACs
- Configuring IPv6 Virtual MACs

Configuring IPv4 Virtual MACs

When you create a IPv4 virtual MAC address and bind it to a interface, any IPv4 packet sent from the interface uses the virtual MAC address that is bound to the interface. If there is no IPv4 virtual MAC bound to an interface, the interface's physical MAC address is used.

The generic virtual MAC is of the form `00:00:5e:00:01:<VRID>`. For example, if you create a VRID with a value of 60 and bind it to an interface, the resulting virtual MAC is 00:00:5e:00:01:3c, where 3c is the hex representation of the VRID. You can create 255 VRIDs with values from 1 to 255.

Creating or modifying an IPv4 Virtual MAC

You create an IPv4 virtual MAC by assigning it a virtual router ID. You can then you bind the virtual MAC to an interface. You cannot bind multiple VRIDs to the same interface. To verify the virtual MAC configuration, you should display and examine the Virtual MACs and the interfaces bound to the Virtual MACs.

To add a virtual MAC by using the command line interface

At the command prompt, type:

- `add vrID`
- `bind vrid <id> -ifnum <interface_name>`
- `show vrID`

Example

```
> add vrID 100
 Done
> bind vrid 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

To unbind interfaces from a virtual MAC by using the command line interface

At the command prompt, type:

- `unbind vrid <id> -ifnum <interface_name>`
- `show vrID`

To configure a virtual MAC by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC** tab, add a new Virtual MAC, or edit an existing Virtual MAC.

Removing an IPv4 Virtual MAC

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove an IPv4 virtual MAC by using the command line interface

At the command prompt, type:

`rm vrid <id>`

Example

```
rm vrid 100s
<!--NeedCopy-->
```

To remove an IPv4 virtual MAC by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC** tab, delete the IPv4 Virtual MAC.

Configuring IPv6 Virtual MACs

The NetScaler supports Vitrtual MAC6 for IPv6 packets. You can bind any interface to a Virtual MAC6, even if an IPv4 virtual MAC is bound to the interface. Any IPv6 packet sent from the interface uses the Vitrtual MAC6 bound to that interface. If there is no Vitrtual MAC6 bound to an interface, an IPv6 packet uses the physical MAC.

Creating or Modifying a Virtual MAC6

You create an IPv6 virtual MAC by assigning it an IPv6 virtual router ID. You can then you bind the virtual MAC to an interface. You cannot bind multiple IPv6 VRIDs to an interface. To verify the Vitrtual MAC6 configuration, you should display and examine the Virtual MAC6s and the interfaces bound to the Virtual MAC6s.

To add a Vitrtual MAC6 by using the command line interface

At the command prompt, type:

- `add vrID6 <id>`
- `bind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

Example

```
> add vrID6 100
Done
> bind vrID6 100 -ifnum 1/1 1/2 1/3
 Done
<!--NeedCopy-->
```

To unbind interfaces from a Vitrtual MAC6 by using the command line interface

At the command prompt, type:

- `unbind vrID6 <id> -ifnum <interface_name>`
- `show vrID6`

To configure a Vitrtual MAC6 by using the GUI

Navigate to **System** > **Network** > **VMAC** and, on the **VMAC6** tab, add a new Virtual MAC6, or edit an existing Virtual MAC6.

Removing a Virtual MAC6

To remove an IPv4 virtual MAC, you delete its virtual router ID.

To remove a Vitrtual MAC6 by using the command line interface

At the command prompt, type:

`rm vrid6 <id>`

Example

```
rm vrid6 100s
<!--NeedCopy-->
```

To remove a Vitrtual MAC6 by using the GUI

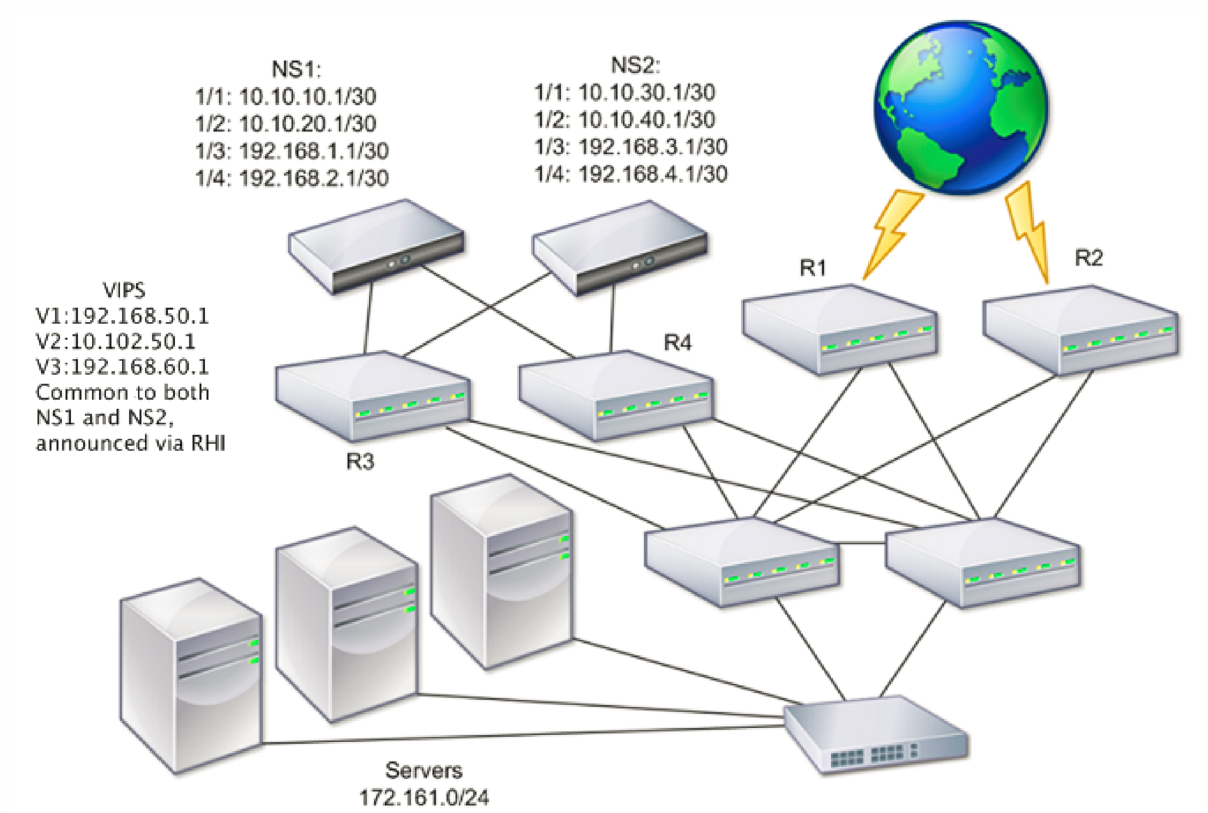
Navigate to **System** > **Network** > **VMAC** and, on the **VMAC6** tab, delete the virtual router ID.

Configuring high availability nodes in different subnets

September 21, 2020

The following figure shows an HA deployment with the two systems located in different subnets:

Figure 1. High Availability over a Routed Network


NS1:
1/1: 10.10.10.1/30
1/2: 10.10.20.1/30
1/3: 192.168.1.1/30
1/4: 192.168.2.1/30
NS2:
1/1: 10.10.30.1/30
1/2: 10.10.40.1/30
1/3: 192.168.3.1/30
1/4: 192.168.4.1/30
VIPS
V1:192.168.50.1
V2:10.102.50.1
V3:192.168.60.1
Common to both NS1 and NS2, announced via RHI
R1
R2
R3
R4
Servers
172.161.0/24

In the figure, the systems NS1 and NS2 are connected to two separate routers, R3 and R4, on two different subnets. The NetScaler appliances exchange heartbeat packets through the routers. This configuration could be expanded to accommodate deployments involving any number of interfaces.

> **Note**:
>
> If you use static routing on your network, you must add static routes between all the systems to ensure that heartbeat packets are sent and received successfully. (If you use dynamic routing on your systems, static routes are unnecessary.)

If the nodes in an HA pair reside on two separate networks, the primary and secondary node must have independent network configurations. This means that nodes on different networks cannot share entities such as SNIPs, VLANs, and routes. This type of configuration, where the nodes in an HA pair have different configurable parameters, is known as Independent Network Configuration (INC) or Symmetric Network Configuration (SNC).

The following table summarizes the configurable entities and options for an INC, and shows how they must be set on each node.

NetScaler entities	Options
IPs (NSIP/SNIPs)	Node-specific. Active only on that node.
VIPs	Floating.

NetScaler entities	Options
VLANs	Node-specific. Active only on that node.
Routes	Node-specific. Active only on that node. Link load balancing routes are floating.
ACLs	Floating (Common). Active on both nodes.
Dynamic routing	Node-specific. Active only on that node. The secondary node should also run the routing protocols and peer with upstream routers.
L2 mode	Floating (Common). Active on both nodes.
L3 mode	Floating (Common). Active on both nodes.
Reverse NAT (RNAT)	Node-specific. RNAT with VIP, because NATIP is floating.

As in configuring HA nodes in the same subnet, to configure HA nodes in different subnets, you log on to each of the two NetScaler appliances and add a remote node representing the other appliance.

Adding a Remote Node

When two nodes of an HA pair reside on different subnets, each node must have a different network configuration. Therefore, to configure two independent systems to function as an HA pair, you must specify INC mode during the configuration process.

When you add an HA node, you must disable the HA monitor for each interface that is not connected or not being used for traffic. For CLI users, this is a separate procedure.

To add a node by using the command line interface

At the command prompt, type:

- `add ha node <id> <IPAddress> -inc ENABLED`
- `show ha node`

Example

```
> add ha node 3 10.102.29.170 -inc ENABLED
 Done
> add ha node 3 1000:0000:0000:0000:0005:0600:700a:888b
  Done
<!--NeedCopy-->
```

To disable an HA monitor by using the command line interface

At the command prompt, type:

- `set interface <ifNum> [-haMonitor ( ON | OFF )]`
- `show interface <ifNum>`

Example

```
> set interface 1/3 -haMonitor OFF
 Done
<!--NeedCopy-->
```

To add a remote node by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, add a new remote node.
2. Make sure to select the Turn off HA monitor on interfaces/channels that are down and Turn on INC (Independent Network Configuration) mode on self mode options.

Removing a Node

If you remove a node, the nodes are no longer in high availability configuration.

To remove a node by using the command line interface

At the command prompt, type:

`rm ha node <id>`

Example

```
> rm ha node 2
  Done
<!--NeedCopy-->
```

To remove a node by using the GUI

Navigate to **System** > **High Availability** and, on the **Nodes** tab, delete the node.

> **Note**:
>
> You can use the Network Visualizer to view the NetScaler appliances that are configured as a high availability (HA) pair and perform high availability configuration tasks.

Configuring route monitors

September 21, 2020

You can use route monitors to make the HA state dependent on the internal routing table, whether or not the table contains any dynamically learned or static routes. In an HA configuration, a route monitor on each node watches the internal routing table to make sure that a route entry for reaching a particular network is always present. If the route entry is not present, the state of the route monitor changes to DOWN.

When a NetScaler appliance has only static routes for reaching a network, and you want to create a route monitor for the network, you must enable monitored static routes (MSR) for the static routes. MSR removes unreachable static routes from the internal routing table. If MSR is disabled on static routes, an unreachable static route can remain in the internal routing table, defeating the purpose of having the route monitor.

Route Monitors are supported both in non-INC and INC mode.

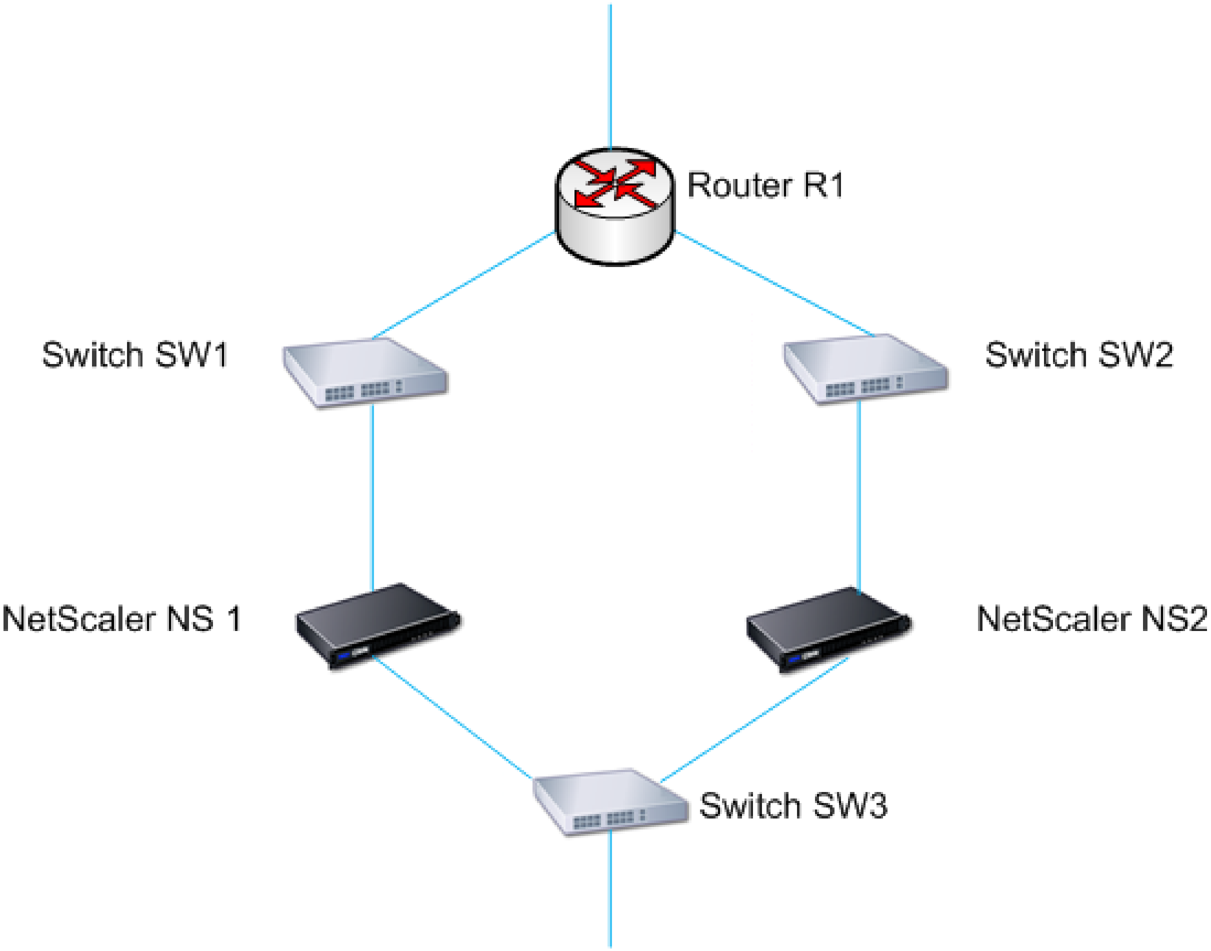
Route Monitors in HA in non-INC mode	Route Monitors in HA in INC mode
Route monitors are propagated by nodes and exchanged during synchronization.	Route monitors are neither propagated by nodes nor exchanged during synchronization.
Route monitors are active only in the current primary node.	Route monitors are active on both the primary and the secondary node.
The NetScaler appliance always displays the state of a route monitor as UP irrespective of the whether the route entry is present or not in the internal routing table.	The NetScaler appliance displays the state of the route monitor as DOWN if the corresponding route entry is not present in the internal routing table.
A route monitor starts monitoring its route after 180 seconds in the following cases [This is done to allow dynamic routes to get learnt, which may take 180 secs]: reboot, failover, set route6 command for v6 routes, set route msr enable/disable command for v4 routes, adding a new route monitor.	-

Route monitors are useful in a non-INC mode HA configuration where you want the non-reachability of a gateway from a primary node to be one of the conditions for HA failover.

Consider an example of a non-Inc mode HA setup in a two-arm topology that has NetScaler appliances NS1 and NS2 in the same subnet, with router R1 and switches SW1, SW2, and SW3.

Because R1 is the only router in this setup, you want the HA setup to failover whenever R1 is not reachable from the current primary node. You can configure a route monitor (say, RM1 and RM2, respectively) on each of the nodes to monitor the reachability of R1 from that node.

Figure 1.



With NS1 as the current primary node, the execution flow is as follows:

1. Route monitor RM1 on NS1 monitors NS1's internal routing table for the presence of a route entry for router R1. NS1 and NS2 exchange heartbeat messages through switch SW1 or SW3 at regular intervals.
2. If switch SW1 goes down, the routing protocol on NS1 detects that R1 is not reachable and therefore removes the route entry for R1 from the internal routing table. NS1 and NS2 exchanges heartbeat messages through switch SW3 at regular intervals.
3. Detecting that the route entry for R1 is not present in the internal routing table, RM1 initiates a failover. If route to R1 is down from both NS1 and NS2, failover happens every 180 seconds till one of the appliances is able to reach R1 and restore the connectivity.

Adding a route monitor to a high availability node

A single procedure creates a route monitor and binds it to an HA node.

To add a route monitor by using the command line interface

At the command prompt, type:

- `bind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- `show HA node`

Example

```
> bind HA node 0 -routeMonitor 10.102.71.0 255.255.255.0
 Done
> bind HA node 0 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
 Done
<!--NeedCopy-->
```

To add a route monitor by using the GUI

Navigate to **System** > **High Availability** and, on the **Route Monitors** tab, click **Configure**.

Removing route monitors

To remove a route monitor by using the command line interface

At the command prompt, type:

- `unbind HA node <id> (-routeMonitor <ip_addr|ipv6_addr> [<netmask>])`
- show ha node

Example

```
unbind HA node 3 -routeMonitor 10.102.71.0 255.255.255.0
unbind HA node 3 -routeMonitor 1000:0000:0000:0000:0005:0600:700a:888b
<!--NeedCopy-->
```

To remove a route monitor by using the GUI

Navigate to **System** > **High Availability** and, on the **Route Monitors** tab, delete the route monitor.

Limiting failovers caused by route monitors in non-INC mode

September 21, 2020

In an HA configuration in non-INC mode, if route monitors fail on both nodes, failover happens every 180 seconds until one of the nodes is able to reach all of the routes monitored by the respective route monitors.

However, for a node, you can limit the number of failovers for a given interval by setting the Maximum Number of Flips and Maximum Flip Time parameters on the nodes. When either limit is reached, no more failovers occur, and the node is assigned as primary (but node state as NOT UP) even if any route monitor fails on that node. This combination of HA state as Primary and Node state as NOT UP is called Stick Primary state.

If the node is then able to reach all of the monitored routes, the next monitor failure triggers resetting of the Maximum Number of Flips and Maximum Flip Time parameters on the node and starting the time specified in the Maximum Flip Time parameter.

These parameters are set independently on each node and therefore are neither propagated nor synchronized.

- Parameters for limiting the number of failovers

  Maximum Number of Flips (maxFlips)

  Maximum number of failovers allowed, within the Maximum Flip Time interval, for the node in HA in non INC mode, if the failovers are caused by route-monitor failure.

  Maximum Flip Time (maxFlipTime)

  Amount of time, in seconds, during which failovers resulting from route-monitor failure are allowed for the node in HA in non INC mode.

To limit the number of failovers by using the command line interface

At the command prompt, type:

- `set HA node [-maxFlips < positive_integer>] [-maxFlipTime <positive_integer >]`
- `show HA node [< id>]`

To limit the number of failovers by using the GUI

1. Navigate to **System** > **High Availability** and, on the **Nodes** tab, open the local node.
2. Set the following parameters:

- Maximum Number of Flips
- Maximum Flip Time

```
    > set ha node -maxFlips 30 -maxFlipTime 60
    Done
    > sh ha node
    1) Node ID: 0
    IP: 10.102.169.82 (NS)
    Node State: UP
    Master State: Primary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: ENABLED
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen :None
    Interfaces causing Partial Failure:None
    SSL Card Status: NOT PRESENT
    Hello Interval: 200 msecs
    Dead Interval: 3 secs
    Node in this Master State for: 0:4:24:1 (days:hrs:min:sec)

    2) Node ID: 1
    IP: 10.102.169.81
    Node State: UP
    Master State: Secondary
    Fail-Safe Mode: OFF
    INC State: DISABLED
    Sync State: SUCCESS
    Propagation: ENABLED
    Enabled Interfaces : 1/1
    Disabled Interfaces : None
    HA MON ON Interfaces : 1/1
    Interfaces on which heartbeats are not seen : None
    Interfaces causing Partial Failure: None
    SSL Card Status: NOT PRESENT

    Local node information:
    Configured/Completed Flips: 30/0
    Configured Flip Time: 60
    Critical Interfaces: 1/1

    Done
<!--NeedCopy-->
```

SNMP Alarm for Sticky Primary State

Enable HA-STICKY-PRIMARY SNMP alarm in a node of a high availability set up if you want to be alerted of the node becoming sticky primary. When the node becomes sticky primary, it alerts by generating a trap message (stickyPrimary (1.3.6.1.4.1.5951.1.1.0.138)) and sends it to all the configured SNMP trap destinations. For more information about configuring SNMP alarms and trap destinations, see onfiguring the NetScaler to Generate SNMPv1 and SNMPv2 Traps.

Frequently Asked Questions

Consider an example of a high availability setup of two NetScaler appliances NS-1 and NS-2 in non-INC mode. Maximum numbers of flips and maximum flip time in both the nodes have been set with the same values.

The following table lists the settings used in this example:

Entity	Detail
IP address of NS-1	10.102.173.211
IP address of NS-2	10.102.173.212
Maximum number of flips	2
Maximum flip time	200

For information about the maximum number of flips and maximum flip time settings, refer to the pdf.

Configuring failover interface set

September 21, 2020

A Failover Interface Set (FIS) is a logical group of interfaces. In an HA configuration, using a FIS is a way to prevent failover by grouping interfaces so that, when one interface fails, other functioning interfaces are still available. A FIS can also be configured for the nodes of a NetScaler cluster.

HA MON interfaces that are not bound to an FIS are known as critical interfaces (CI) because if any of them fails, failover is triggered.

> **Note**:
>
> An FIS does not create an active and standby configuration. It also does not prevent bridging loops when connecting to links to the same VLAN.

> **Note**
>
> An FIS does not create an active and standby Interfaces or channels. It also does not prevent bridging loops when connecting to links to the same VLAN

Creating or Modifying an FIS

To add an FIS and bind interfaces to it by using the command line interface

At the command prompt, type:

- `add fis <name>`
- `bind fis <name> <ifnum> ...`
- `show fis <name>`

Example

```
> add fis fis1
 Done
> bind fis fis1 1/3 1/5
 Done
<!--NeedCopy-->
```

An unbound interface becomes a critical interface (CI) if it is enabled and HA MON is on.

To unbind an interface from an FIS by using the command line interface

At the command prompt, type:

- `unbind fis <name> <ifnum> ...`
- `show fis <name>`

Example

```
> unbind fis fis1 1/3
 Done
<!--NeedCopy-->
```

To configure an FIS by using the GUI

Navigate to System > High Availability and, on the Failover Interface Set tab, add a new FIS, or edit an existing FIS.

Removing an FIS

When the FIS is removed, its interfaces are marked as critical interfaces.

To remove an FIS by using the command line interface

At the command prompt, type:

`rm fis <name>`

Example

```
> rm fis fis1
  Done
<!--NeedCopy-->
```

To remove an FIS by using the GUI

Navigate to **System** > **High Availability** and, on the **Failover Interface Set** tab, delete the FIS.

Understanding the causes of failover

August 18, 2021

The following events can cause failover in an high availability configuration:

1. If the secondary node does not receive a heartbeat packet from the primary for a period of time that exceeds the dead interval set on the secondary. (See Note 1.)
2. The primary node experiences a hardware failure of its SSL card.
3. The primary node does not receive any heartbeat packets on its network interfaces for three seconds.
4. On the primary node, a network interface that is not part of a Failover Interface Set (FIS) or a Link Aggregation (LA) channel and has the HA Monitor (HAMON) enabled, fails. (See Note 2.)
5. On the primary node, all interfaces in an FIS fail. (See Note 2.)
6. On the primary node, an LA channel with HAMON enabled fails. (See Note 2.)
7. On the primary node, all interfaces fail (see Note 2). In this case, failover occurs regardless of the HAMON configuration.
8. On the primary node, all interfaces are manually disabled. In this case, failover occurs regardless of the HAMON configuration.
9. You force a failover by issuing the force failover command on either node.
10. A route monitor that is bound to the primary node goes DOWN.

> **Note 1:**
>
> For more information about setting the dead interval, see Configuring the Communication Intervals. Possible causes for a node not receiving heartbeat packets from a peer node include:

> - A network configuration problem prevents heartbeats from traversing the network between the HA nodes.
> - The peer node experiences a hardware or software failure that causes it to freeze (hang), reboot, or otherwise stop processing and forwarding heartbeat packets.
>
> **Note 2:**
>
> In this case, fail means that the interface was enabled but goes to the DOWN state, as can be seen from the show interface command or from the GUI. Possible causes for an enabled interface to be in the DOWN state are LINK DOWN and TXSTALL.

Forcing a node to fail over

June 29, 2021

You might want to force a failover if, for example, you need to replace or upgrade the primary node. You can force failover from either the primary or the secondary node. A forced failover is not propagated or synchronized. To view the synchronization status after a forced failover, you can view the status of the node.

A forced failover fails in any of the following circumstances:

- You force failover on a standalone system.
- The secondary node is disabled.
- The secondary node is configured to remain secondary.

The NetScaler appliance displays a warning message if it detects a potential issue when you run the force failover command. The message includes the information that triggered the warning, and requests confirmation before proceeding.

You can force a failover on a primary node, secondary node, and when nodes are in listen mode.

- **Forcing Failover on the Primary Node**.

  If you force failover on the primary node, the primary becomes the secondary and the secondary becomes the primary. Forced failover is possible only when the primary node can determine that the secondary node is UP.

  If the secondary node is DOWN, the force failover command returns the following error message: "Operation not possible due to invalid peer state. Rectify and retry."

  If the secondary system is in the claiming state or inactive, it returns the following error message:

  ```
  Operation not possible now. Please wait for the system to stabilize
  before retrying.
  ```

- **Forcing Failover on the Secondary Node**.

  If you run the force failover command from the secondary node, the secondary node becomes primary and the primary node becomes secondary. A force failover can occur only if the secondary node's health is good and it is not configured to stay secondary.

  If the secondary node cannot become the primary node, or if the secondary node was configured to stay secondary (using the STAYSECONDARY option), the node displays the following error message:

  ```
  Operation not possible as my state is invalid. View the node for more
  information.
  ```

- **Forcing Failover When Nodes Are in Listen Mode**.

  When the two nodes of an HA pair are running different versions of the system software, the node running the higher version switches to the listen mode. In this mode, neither command propagation nor synchronization works.

  Before upgrading the system software on both nodes, test the new version on one of the nodes. To do this, you must force a failover on the system that has already been upgraded. The upgraded system then takes over as the primary node, but neither command propagation or synchronization occurs. Also, all connections must be re-established.

> **Important!**
>
> If you force a failover when an HA synchronization operation is in progress, some active data sessions on the HA setup might be lost. So, wait for the HA synchronization operation to be completed before performing the force failover operation.

To force failover on a node by using the command line interface:

At the command prompt, type:

```
force HA failover
```

To force failover on a node by using the GUI:

Navigate to **System** > **High Availability** and, on the **Nodes** tab, select the node, in the Action list, select **Force Failover**.

Forcing the secondary node to stay secondary

October 14, 2020

In an HA setup, the secondary node can be forced to stay secondary regardless of the state of the primary node.

For example, suppose the primary node needs to be upgraded and the process will take a few seconds. During the upgrade, the primary node may go down for a few seconds, but you do not want the secondary node to take over; you want it to remain the secondary node even if it detects a failure in the primary node.

When you force the secondary node to stay secondary, it will remain secondary even if the primary node goes down. Also, when you force the status of a node in an HA pair to stay secondary, it does not participate in HA state machine transitions. The status of the node is displayed as STAYSECONDARY.

Forcing the node to stay secondary works on both standalone and secondary nodes. On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node stops processing traffic and becomes the secondary node. The new node becomes the primary node.

> **Note**:
>
> When you force a system to remain secondary, the forcing process is not propagated or synchronized. It affects only the node on which you run the command.

To force the secondary node to stay secondary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYSECONDARY
```

To force the secondary node to stay secondary by using the GUI

Navigate to **System** > **High Availability**, on the **Nodes** tab, open the local node, and select **STAY SECONDARY**.

Forcing the primary node to stay primary

May 18, 2021

In an HA setup, you can force a healthy primary node to remain primary even after a failover. You can enable this option either on a primary node in an HA pair. This option allows the primary node to be in primary state as long as it is healthy.

On a standalone node, you must use this option before you can add a node to create an HA pair. When you add the new node, the existing node continues to function as the primary node, and the new node becomes the secondary node.

To force the primary node to stay primary by using the command line interface

At the command prompt, type:

```
set ha node -hastatus STAYPRIMARY
```

To force the primary node to stay primary by using the GUI

Navigate to **System** > **High Availability**, on the **Nodes** tab, open the local node, and select **STAY PRIMARY**.

Understanding the high availability health check computation

September 21, 2020

The following table summarizes the factors examined in a health check computation:

- State of the failover interface sets
- State of the critical interfaces
- State of the route monitors

The following table summarizes the health check computation.

Failover interface sets	Critical interfaces	Route monitor	Condition
N	Y	N	If the system has any critical interfaces, all of those critical interfaces must be UP.
Y	Y	N	If the system has any failover interface sets, all of those failover interface sets must be UP.
Y	Y	Y	If the system has any route monitors configured, all monitored routes must be present in the failover interface set.

High availability FAQ

October 13, 2020

1. What are the various ports used to exchange the HA-related information between the nodes in an HA configuration?

   In an HA configuration, both nodes use the following ports to exchange HA related information:

   - UDP Port 3003, to exchange heartbeat packets.
   - Port 3010, for synchronization and command propagation.

2. What are the conditions that trigger synchronization?

   Synchronization is triggered by any of the following conditions:

   - The incarnation number of the primary node, received by the secondary, does not match that of the secondary node.
     Note: Both nodes in an HA configuration maintain a counter called *incarnation number*, which counts the number of configurations in the node's configuration file. Each node sends its incarnation number to each other node in the heartbeat messages. The incarnation number is not incremented for the following commands:
     a) All HA configuration related commands. For example, add ha node, set ha node, and bind ha node.
     b) All Interface related commands. For example, set interface and unset interface.
     c) All channel-related commands. For example, add channel, set channel, and bind channel.
   - The secondary node comes up after a restart.
   - The primary node becomes secondary after a failover.

3. What configurations are not synced or propagated in an HA configuration in INC or non-INC mode?

   The following commands are neither propagated nor synced to the secondary node:

   - All node specific HA configuration commands. For example, add ha node, set ha node, and bind ha node.
   - All Interface related configuration commands. For example, set interface and unset interface.
   - All channel related configuration commands. For example, add channel, set channel, and bind channel.

   > **Note:**
   >
   > The following configurations are neither synced nor propagated only in HA in INC mode. Each node has its own:

> - SNIPs
> - VLANs
> - Routes (except LLB routes)
> - Route monitors
> - RNAT rules (except any RNAT rule with VIP as the NAT IP)
> - Dynamic routing configurations
> - Net profiles

4. Does a configuration added to the secondary node get synchronized on the primary?

   No, a configuration added to the secondary node is not synchronized to the primary.

5. What could be the reason for both nodes claiming to be the primary in an HA configuration?

   The most likely reason is that the primary and secondary nodes are both healthy but the secondary does not receive the heartbeat packets from the primary. The problem could be with the network between the nodes.

6. Does an HA configuration run into any issues if you deploy the two nodes with different system clock settings?

   Different system-clock settings on the two nodes can cause the following issues:

   - The time stamps in the log file entries do not match. This situation makes it difficult to analyze the log entries for any issues.
   - After a failover, you might have problems with any type of cookie based persistence for load balancing. A significant difference between the times can cause a cookie to expire sooner than expected, resulting in termination of the persistence session.
   - Similar considerations apply to any time related decisions on the nodes.

7. What are the conditions for failure of the
   force HA sync command?

   Forced synchronization fails in any of the following circumstances:

   - You force synchronization when synchronization is already in progress.
   - You force synchronization on a standalone NetScaler appliance.
   - The secondary node is disabled.
   - HA synchronization is disabled on the current secondary node.
   - HA propagation is disabled on the current primary node and you force synchronization from the primary.

8. What are the conditions for failure of the
   `sync HA files command`?

   Synchronizing configuration files fail in either of the following circumstances:

   - On a standalone system.

  - With the secondary node disabled.

9. In an HA configuration, if the secondary node takes over as the primary, does it switch back to secondary status if the original primary comes back online?

   No. After the secondary node takes over as the primary, it remains as primary even if the original primary node comes back online again. To interchange the primary and secondary status of the nodes, run the `force failover` command.

10. What are the conditions for failure of the `force failover command`?

    A forced failover fails in any of the following circumstances:

    - You force failover on a standalone system.
    - The secondary node is disabled.
    - The secondary node is configured to remain secondary.
    - The primary node is configured to remain primary.
    - The state of the peer node is unknown.

11. What types of traffic flows between the nodes of a high availability set-up?

    In a high availability set-up, traffic related to maintaining the high availability configuration flows between the two nodes. This traffic is of the following types:

    - Config synchronization
    - Config propagation
    - Data connection mirroring (For features that have connection failover enabled. For more information see, Connection failover.)
    - Load balancing persistent session synchronization
    - Session state synchronization
    - Session state ICA sessions (If ICA session reliability is enabled. For more information, see Session reliability.)

Troubleshooting high availability issues

September 21, 2020

The most common high availability issues involve the high availability feature not working at all, or working only intermittently. Following are common high availability issues, and probable causes and resolutions.

- **Issue**
  The inability of the NetScaler appliances to pair the NetScaler appliances in a high availability setup.
  - **Cause**
    Network connectivity
    Resolution
    Verify that both the appliances are connected to the switch and the interfaces are enabled.
  - **Cause**
    Mismatch in the Password for the default Administrator account
    Resolution
    Verify that the password on both the appliances is the same.
  - **Cause**
    IP conflict
    Resolution
    Verify that both the appliances have unique NetScaler IP (NSIP) address. The appliances should not have the same NSIP address.
  - **Cause**
    Node ID mismatch
    Resolution
    Verify that the Node ID Configuration on both the appliances is unique. The appliances should not have the same Node ID configuration. Additionally, you must assign value for a Node ID between 1 and 64.
  - **Cause**
    Mismatch in the password of the RPC node
    Resolution
    Verify that both the nodes have the same RPC node password.
  - **Cause**
    An administrator has disabled the remote node
    Resolution
    Enable the remote node.
  - **Cause**
    The Firewall application has blocked the heartbeat packets
    Resolution

Verify that the UDP port 3003 is allowed.

- **Issue**
  Both the appliances claim to be the primary appliance.
  - **Cause**
    Missing heartbeat packets between the appliances
    Resolution

    Verify that the UDP port 3003 is not blocked for communication between the appliances.

- **Issue**

  The NetScaler appliance is not able to synchronize the configuration.

  - **Cause**

    A Firewall application is blocking the required port.

    Resolution

    Verify that the UDP port 3010 (or UDP port 3008 with secure synchronization) is not blocked for communication between the appliances.

  - **Cause**

    An administrator has disabled synchronization.

    Resolution

    Enable synchronization on the appliance that has the issue.

  - **Cause**

    Different NetScaler releases or builds are installed on appliances.

    Resolution

    Upgrade the appliances to the same NetScaler release or build.

- **Issue**

  Command propagation fails between the appliances.

  - **Cause**

    A Firewall application is blocking the port.

    Resolution

    Verify that the UDP port 3011 (or UDP port 3009 with secure propagation) is not blocked for communication between the appliances.

  - **Cause**

    An administrator has disabled command propagation.

    Resolution

    Enable command propagation on the appliance that has the issue.

  - **Cause**

    Different NetScaler releases or builds are installed on appliances.

    Resolution

    Upgrade the appliances to the same NetScaler release or build.

- **Issue**

  The NetScaler appliances in the high availability pair are unable to run the force failover process.

  - **Cause**

    The Secondary node is disabled.

    Resolution

    Enable the secondary node.

  - **Cause**

    The Secondary node is configured to stay secondary.

    Resolution
    Set the secondary high availability status of the secondary node to Enable from Stay Secondary.

- **Issue**
  The secondary appliance does not receive any traffic after the failover process.
  - **Cause**
    The upstream router does not understand GARP messages of NetScaler appliance.
    Resolution
    Configure Virtual MAC address on the secondary appliance.

Managing high availability heartbeat messages on a NetScaler appliance

September 21, 2020

The two nodes in a high availability configuration send and receive heartbeat messages to and from each other on all interfaces that are enabled. The heartbeat messages flow regardless of the HA MON setting on these interfaces. If NSVLAN or both (NSVLAN and SYNC) are configured on an appliance, the heartbeat messages flow only through the enabled interfaces that are part of the NSVLAN and SYNCVLAN.

If a node does not receive the heartbeat messages on an enabled interface, it sends critical alerts to the specified Command Center and SNMP managers. These critical alerts give false alarms and draw unnecessary attention from the administrators for interfaces that are not configured as part of the connections to the peer node.

To resolve this issue, the HAHeartBeat option for interfaces and channels is used for enabling or disabling HA heartbeat-message flow on them.

To manage the high availability heartbeat messages on an interface by using the command line interface

At the command prompt, type:

- `set interface <ID> [-HAHeartBeat ( ON | OFF )]`
- `show interface <ID>`

To manage the high availability heartbeat messages on a channel by using the command line interface

At the command prompt, type:

- `set channel <ID> [-HAHeartBeat ( ON | OFF )]`
- `show channel <ID>`

To manage the high availability heartbeat messages for an interface by using the GUI

1. Navigate to **System** > **Network** > **Interfaces**.
2. Enable or disable the **HA Heart Beat** parameter.

To manage the high availability heartbeat messages on a channel by using the GUI

1. Navigate to **System** > **Network** > **Channels**.
2. Enable or disable the **HA Heart Beat** parameter.

TCP optimization

September 21, 2020

TCP uses the following optimization techniques and congestion control strategies (or algorithms) to avoid network congestion in data transmission.

Congestion Control Strategies

The Transmission Control Protocol (TCP) has long been used to establish and manage Internet connections, handle transmission errors, and smoothly connect web applications with client devices. But network traffic has become more difficult to control, because packet loss does not depend only on the congestion in the network, and congestion does not necessarily cause packet loss. Therefore, to measure congestion, a TCP algorithm should focus on both packet loss and bandwidth.

Proportional Rate Recovery (PRR) Algorithm

TCP Fast Recovery mechanisms reduce web latency caused by packet losses. The new Proportional Rate Recovery (PRR) algorithm is a fast recovery algorithm that evaluates TCP data during a loss recovery. It is patterned after Rate-Halving, by using the fraction that is appropriate for the target window chosen by the congestion control algorithm. It minimizes window adjustment, and the actual window size at the end of recovery is close to the Slow-Start threshold (ssthresh).

TCP Fast Open (TFO)

TCP Fast Open (TFO) is a TCP mechanism that enables speedy and safe data exchange between a client and a server during TCP's initial handshake. This feature is available as a TCP option in the TCP profile bound to a virtual server of a NetScaler appliance. TFO uses a TCP Fast Open Cookie (a security cookie) that the NetScaler appliance generates to validate and authenticate the client initiating a TFO connection to the virtual server. By using the TFO mechanism, you can reduce an application's network latency by the time required for one full round trip, which significantly reduces the delay experienced in short TCP transfers.

How TFO works

When a client tries to establish a TFO connection, it includes a TCP Fast Open Cookie with the initial SYN segment to authenticate itself. If authentication is successful, the virtual server on the NetScaler appliance can include data in the SYN-ACK segment even though it has not received the final ACK segment of the three-way handshake. This saves up to one full round-trip compared to a normal TCP connection, which requires a three-way handshake before any data can be exchanged.

A client and a backend server perform the following steps to establish a TFO connection and exchange data securely during the initial TCP handshake.

1. If the client does not have a TCP Fast Open Cookie to authenticate itself, it sends a Fast Open Cookie request in the SYN packet to the virtual server on the NetScaler appliance.
2. If the TFO option is enabled in the TCP profile bound to the virtual server, the appliance generates a cookie (by encrypting the client's IP address under a secret key) and responds to the client with a SYN-ACK that includes the generated Fast Open Cookie in a TCP option field.
3. The client caches the cookie for future TFO connections to the same virtual server on the appliance.
4. When the client tries to establish a TFO connection to the same virtual server, it sends SYN that includes the cached Fast Open Cookie (as a TCP option) along with HTTP data.
5. The NetScaler appliance validates the cookie, and if the authentication is successful, the server accepts the data in the SYN packet and acknowledges the event with a SYN-ACK, TFO Cookie, and HTTP Response.

> **Note:**
>
> If the client authentication fails, the server drops the data and acknowledges the event only with a SYN indicating a session timeout.

1. On the server side, if the TFO option is enabled in a TCP profile bound to a service, the NetScaler appliance determines whether the TCP Fast Open Cookie is present in the service to which it is trying to connect.
2. If the TCP Fast Open Cookie is not present, the appliance sends a cookie request in the SYN packet.
3. When the backend server sends the Cookie, the appliance stores the cookie in the server information cache.
4. If the appliance already has a cookie for the given destination IP pair, it replaces the old cookie with the new one.
5. If the cookie is available in the server information cache when the virtual server tries to reconnect to the same backend server by using the same SNIP address, the appliance combines the

data in SYN packet with the cookie and sends it to the backend server.

6. The backend server acknowledges the event with both data and a SYN.

Note: If the server acknowledges the event with only a SYN segment, the NetScaler appliance immediately resends the data packet after removing the SYN segment and the TCP options from the original packet.

Configuring TCP fast open

To use the TCP Fast Open (TFO) feature, enable the TCP Fast Open option in the relevant TCP profile and set the TFO Cookie Timeout parameter to a value that suits the security requirement for that profile.

To enable or disable TFO by using the command line:

At the command prompt, type one of the following commands to enable or disable TFO in a new or existing profile.

Note: The default value is DISABLED.

```
    add tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
    set tcpprofile <TCP Profile Name> - tcpFastOpen ENABLED | DISABLED
    unset tcpprofile <TCP Profile Name> - tcpFastOpen
    Examples
    add tcpprofile Profile1 - tcpFastOpen
    Set tcpprofile Profile1 - tcpFastOpen Enabled
    unset tcpprofile Profile1 - tcpFastOpen
<!--NeedCopy-->
```

To set TCP Fast Open cookie timeout value by using the command line interface

At the command prompt, type:

```
    set tcpparam - tcpfastOpenCookieTimeout <Timeout Value>
    Example
    set tcpprofile - tcpfastOpenCookieTimeout 30secs
<!--NeedCopy-->
```

To configure the TCP Fast Open by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **TCP Fast Open** checkbox.
3. Click **OK** and then **Done**.

To Configure the TCP Fast Cookie timeout value by using the GUI

Navigate to **Configuration** > **System** > **Settings** > **Change TCP Parameters** and then **Configure TCP Parameters** page to set the TCP Fast Open Cookie timeout value.

TCP Hystart

A new TCP profile parameter, hystart, enables the Hystart algorithm, which is a slow-start algorithm that dynamically determines a safe point at which to terminate (ssthresh). It enables a transition to congestion avoidance without heavy packet losses. This new parameter is disabled by default.

If congestion is detected, Hystart enters a congestion avoidance phase. Enabling it gives you better throughput in high-speed networks with high packet loss. This algorithm helps maintain close to maximum bandwidth while processing transactions. It can therefore improve throughput.

Configuring TCP Hystart

To use the Hystart feature, enable the Cubic Hystart option in the relevant TCP profile.

To configure Hystart by using the command line interface (CLI)

At the command prompt, type one of the following commands to enable or disable Hystart in a new or existing TCP profile.

```
    add tcpprofile <profileName> -hystart ENABLED
    set tcpprofile <profileName> -hystart ENABLED
    unset tcprofile <profileName> -hystart
<!--NeedCopy-->
```

Examples:

```
    add tcpprofile profile1 -hystart ENABLED
    set tcpprofile profile1 -hystart ENABLED
    unset tcprofile profile1 -hystart
<!--NeedCopy-->
```

To configure Hystart support by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select the **Cubic Hystart** check box.
3. Click **OK** and then **Done**.

TCP burst rate control

It is observed that TCP control mechanisms can lead to a traffic burst on high speed mobile networks with a negative impact on the overall network efficiency. Due to mobile network conditions such as congestion or Layer-2 retransmission of data, TCP acknowledgements arrive clumped at the sender triggering a burst of transmission. These group of consecutive packets sent with a short inter-packet gaps it is called TCP packet burst. To overcome the traffic burst, the NetScaler appliance uses a TCP Burst Rate Control technique. This technique evenly spaces data into the network for an entire round-trip-time so that the data is not sent into a burst. By using this burst rate control technique, you can achieve better throughput and lower packet drop rates.

How TCP burst rate control works

In a NetScaler appliance, this technique evenly spreads the transmission of a packets across the entire duration of the round-trip-time (RTT). This is achieved by using a TCP stack and network packet scheduler that identifies the various network conditions to output packets for ongoing TCP sessions in order to reduce the bursts.

At the sender, instead of transmitting packets immediately upon receipt of an acknowledgment, the sender can delay transmitting packets to spread them out at the rate defined by scheduler (Dynamic configuration) or by the TCP profile (Fixed configuration).

Configuring TCP burst rate control

To use the TCP Burst Rate Control option in the relevant TCP profile and set the burst rate control parameters.

To set TCP burst rate control by using the command line

At the command prompt, set one of the following TCP Burst Rate Control commands are configured in a new or existing profile.

Note: The default value is DISABLED.

```
add tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic | Fixed

set tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic | Fixed

unset tcpprofile <TCP Profile Name> -burstRateControl Disabled | Dynamic | Fixed
<!--NeedCopy-->
```

where,

Disabled – If the Burst rate control is disabled, then a NetScaler appliance does not perform burst management other than the maxBurst setting.

Fixed – If the TCP burst rate control is Fixed, the appliance uses the TCP Connection Payload Send Rate value mentioned in the TCP Profile.

Dynamic – If the Burst Rate Control is "Dynamic" the connection is being regulated based on various network conditions to reduce TCP bursts. This mode works only when the TCP connection is in ENDPOINT mode. When Dynamic Burst Rate control is enabled the maxBurst parameter of the TCP profile is not in effect.

```
    add tcpProfile  profile1 -burstRateControl Disabled

    set tcpProfile profile1 -burstRateControl Dynamic

    unset tcpProfile profile1 -burstRateControl Fixed
<!--NeedCopy-->
```

To set TCP Burst Rate Control parameters by using the command line interface

At the command prompt, type:

```
    set ns tcpprofile nstcp_default_profile – burstRateControl <type of burst rate control> – tcprate <TCP rate> -rateqmax <maximum bytes in queue>

    T1300-10-2> show ns tcpprofile nstcp_default_profile
            Name: nstcp_default_profile
            Window Scaling status:  ENABLED
            Window Scaling factor: 8
            SACK status: ENABLED
            MSS: 1460
            MaxBurst setting: 30 MSS
            Initial cwnd setting: 16 MSS
            TCP Delayed-ACK Timer: 100 millisec
            Nagle's Algorithm: DISABLED
            Maximum out-of-order packets to queue: 15000
            Immediate ACK on PUSH packet: ENABLED
            Maximum packets per MSS: 0
            Maximum packets per retransmission: 1
            TCP minimum RTO in millisec: 1000
            TCP Slow start increment: 1
            TCP Buffer Size: 8000000 bytes
            TCP Send Buffer Size: 8000000 bytes
            TCP Syncookie: ENABLED
            Update Last activity on KA Probes: ENABLED
            TCP flavor: BIC
            TCP Dynamic Receive Buffering: DISABLED
            Keep-alive probes: ENABLED
            Connection idle time before starting keep-alive probes: 900 seconds
            Keep-alive probe interval: 75 seconds
            Maximum keep-alive probes to be missed before dropping connection: 3
            Establishing Client Connection: AUTOMATIC
            TCP Segmentation Offload: AUTOMATIC
            TCP Timestamp Option: DISABLED
            RST window attenuation (spoof protection): ENABLED
            Accept RST with last acknowledged sequence number: ENABLED
            SYN spoof protection: ENABLED
            TCP Explicit Congestion Notification: DISABLED
            Multipath TCP: DISABLED
            Multipath TCP drop data on pre-established subflow: DISABLED
            Multipath TCP fastopen: DISABLED
            Multipath TCP session timeout: 0 seconds
            DSACK: ENABLED
            ACK Aggregation: DISABLED
            FRTO: ENABLED
            TCP Max CWND : 4000000 bytes
            FACK: ENABLED
            TCP Optimization mode: ENDPOINT
            TCP Fastopen: DISABLED
            HYSTART: DISABLED
            TCP dupack threshold: 3
            Burst Rate Control: Dynamic
            TCP Rate: 0
            TCP Rate Maximum Queue: 0
<!--NeedCopy-->
```

To configure the TCP Burst Rate Control by using the GUI

1. Navigate to **Configuration** > **System** > **Profiles** > and then click **Edit** to modify a TCP profile.
2. On the **Configure TCP Profile** page, select **TCP Burst Control** option from the drop-down list:
    a) BurstRateCntrl
    b) CreditBytePrms

   c) RateBytePerms
   d) RateSchedulerQ
3. Click **OK** and then **Done**.

Protection against wrapped sequence (PAWS) algorithm

If you enable the TCP timestamp option in the default TCP profile, the NetScaler appliance uses the Protection Against Wrapped Sequence (PAWS) algorithm to identify and reject old packets whose sequence numbers are within the current TCP connection's receive window because the sequence has "wrapped" (reached its maximum value and restarted from 0).

If network congestion delays a non-SYN data packet and you open a new connection before the packet arrives, sequence-number wrapping might cause the new connection to accept the packet as valid, leading to data corruption. But if the TCP timestamp option is enabled, the packet is discarded.

By default, the TCP timestamp option is disabled. If you enable it, the appliance compares the TCP timestamp (SEG.TSval) in a packet's header with the recent timestamp (Ts.recent) value. If SEG.TSval is equal to or greater than Ts.recent, the packet is processed. Otherwise, the appliance drops the packet and sends a corrective acknowledgement.

How PAWS works

The PAWS algorithm processes all the incoming TCP packets of a synchronized connection as follows:

1. If SEG.TSval < Ts.recent: The incoming packet is not acceptable. PAWS sends an acknowledgement (as specified in RFC-793) and drops the packet. Note: Sending an ACK segment is necessary in order to retain TCP's mechanisms for detecting and recovering from half-open connections.
2. If packet is outside the window: PAWS rejects the packet, as in normal TCP processing.
3. If SEG.TSval > Ts.recent: PAWS accepts the packet and processes it.
4. If SEG.TSval <= Last.ACK.sent (arriving segment satisfies): PAWS must copies SEG.TSval value to Ts.recent (is it copied to Ts.Recent field in the db?.
5. If packet is in sequence: PAWS accepts the packet.
6. If packet is not in sequence: The packet is treated as a normal in-window, out-of-sequence TCP segment. For example, it might be queued for later delivery.
7. If Ts.recent value is idle for more than 24 days: The validity of Ts.recent is checked if the PAWS timestamp check fails. If the Ts.recent value is found to be invalid, the segment is accepted and the PAWS rule updates Ts.recent with the TSval value from the new segment.

To enable or disable TCP timestamp by using the command line interface

At the command prompt, type:

```
set nstcpprofile nstcp_default_profile -TimeStamp (ENABLED | DISABLED)
<!--NeedCopy-->
```

To enable or disable TCP timestamp by using the GUI

Navigate to **System** > **Profile** > **TCP Profile**, select the default TCP profile, click **Edit**, and select or clear the **TCP timestamp** check box.

Optimization Techniques

TCP uses the following optimization techniques and methods for optimized flow controls.

Policy based TCP Profile Selection

Network traffic today is more diverse and bandwidth-intensive than ever before. With the increased traffic, the effect that Quality of Service (QoS) has on TCP performance is significant. To enhance QoS, you can now configure AppQoE policies with different TCP profiles for different classes of network traffic. The AppQoE policy classifies a virtual server's traffic to associate a TCP profile optimized for a particular type of traffic, such as 3G, 4G, LAN, or WAN.

To use this feature, create a policy action for each TCP profile, associate an action with AppQoE policies, and bind the policies to the load balancing virtual servers.

For information about using subscriber attributes to perform TCP optimization, see Policy-based TCP Profile.

Configuring Policy Based TCP Profile Selection

Configuring policy based TCP profile selection consists of the following tasks:

- Enabling AppQoE. Before configuring the TCP profile feature, you must enable the AppQoE feature.
- Adding AppQoE Action. After enabling the AppQoE feature, configure an AppQoE action with a TCP profile.
- Configuring AppQoE based TCP Profile Selection. To implement TCP profile selection for different classes of traffic, you must configure AppQoE policies with which your NetScaler ADC can distinguish the connections and bind the correct AppQoE action to each policy.
- Binding AppQoE Policy to Virtual Server. Once you have configured the AppQoE policies, you must bind them to one or more load balancing, content switching, or cache redirection virtual servers.

Configuring using the command line interface

To enable AppQoE by using the command line interface

At the command prompt, type the following commands to enable the feature and verify that it is enabled:

- `enable ns feature appqoe`
- `show ns feature`

To bind a TCP profile while creating an AppQoE action using the command line interface

At the command prompt, type the following AppQoE action command with tcpprofiletobind option.

```
add appqoe action <name> [-priority <priority>] [-respondWith ( ACS | NS ) [<CustomFile>] [-altContentSvcName <string>] [-altContentPath <string>] [-maxConn <positive_integer>] [-delay <usecs>]] [-polqDepth <positive_integer>] [-priqDepth <positive_integer>] [-dosTrigExpression <expression>] [-dosAction ( SimpleResponse |HICResponse )] [-tcpprofiletobind <string>]
show appqoe action
```

To configure an AppQoE policy by using the command line interface

At the command prompt, type:

```
add appqoe policy <name> -rule <expression> -action <string>
```

To bind an AppQoE policy to load balancing, cache redirection or content switching virtual servers by using the command line interface

At the command prompt, type:

```
bind cs vserver cs1 -policyName <appqoe_policy_name> -priority <priority>
bind lb vserver <name> - policyName <appqoe_policy_name> -priority <priority>
bind cr vserver <name> -policyName <appqoe_policy_name> -priority <priority>
```

Example

```
add ns tcpProfile tcp1 -WS ENABLED -SACK ENABLED -WSVal 8 -nagle ENABLED -maxBurst 30 -initialCwnd 16 -oooQSize 15000 -minRTO 500 -slowStartIncr 1 -bufferSize 4194304 -flavor BIC -KA ENABLED -sendBuffsize 4194304 -rstWindowAttenuate ENABLED -spoofSynDrop
```

```
        ENABLED -dsack enabled -frto ENABLED -maxcwnd 4000000 -fack
        ENABLED -tcpmode ENDPOINT
    add appqoe action appact1 -priority HIGH -tcpprofile tcp1
    add appqoe policy apppol1 -rule "client.ip.src.eq(10.102.71.31)" -
        action appact1
    bind lb vserver lb2 -policyName apppol1 -priority 1 -
        gotoPriorityExpression END -type REQUEST
    bind cs vserver cs1 -policyName apppol1 -priority 1 -
        gotoPriorityExpression END -type REQUEST
<!--NeedCopy-->
```

Configuring Policy based TCP Profiling using the GUI

To enable AppQoE by using the GUI

1. Navigate to **System** > **Settings**.
2. In the details pane, click **Configure Advanced Features.**
3. In the **Configure Advanced Features** dialog box, select the **AppQoE** check box.
4. Click **OK**.

To configure AppQoE policy by using the GUI

1. Navigate to **App-Expert** > **AppQoE** > **Actions**.
2. In the details pane, do one of the following:
3. To create a new action, click **Add**.
4. To modify an existing action, select the action, and then click **Edit**.
5. In the **Create AppQoE Action** or the **Configure AppQoE Action** screen, type or select values for the parameters. The contents of the dialog box correspond to the parameters described in "Parameters for configuring the AppQoE Action" as follows (asterisk indicates a required parameter):
    a) Name—name
    b) Action type—respondWith
    c) Priority—priority
    d) Policy Queue Depth—polqDepth
    e) Queue Depth—priqDepth
    f) DOS Action—dosAction
6. Click **Create**.

To bind AppQoE policy by using the GUI

1. Navigate to **Traffic Management** > **Load Balancing** > **Virtual Servers**, select a server and then click **Edit**.
2. In the **Policies** section and click (+) to bind an AppQoE policy.
3. In the **Policies** slider, do the following:
   a) Select a policy type as AppQoE from the drop-down list.
   b) Select a traffic type from the drop-down list.
4. In the **Policy Binding** section, do the following:
   a) Click **New** to create a new AppQoE policy.
   b) Click **Existing Policy** to select an AppQoE policy from the drop-down list.
5. Set the binding priority and click **Bind** to the policy to the virtual server.
6. Click **Done**.

SACK block generation

TCP performance slows down when multiple packets are lost in one window of data. In such a scenario, a Selective Acknowledgement (SACK) mechanism combined with a selective repeat retransmission policy overcomes this limitation. For every incoming out-of-order packet, you must generate a SACK block.

If the out-of-order packet fits in the reassembly queue block, insert packet info in the block, and set the complete block info as SACK-0. If an out-of-order packet does not fit into reassembly block, send the packet as SACK-0 and repeat the earlier SACK blocks. If an out-of-order packet is a duplicate and packet info is set as SACK-0 then D-SACK the block.

Note: A packet is considered as D-SACK if it is an acknowledged packet, or an out of order packet which is already received.

Client reneging

A NetScaler appliance can handle client reneging during SACK based recovery.

Memory checks for marking end_point on PCB is not considering total available memory

In a NetScaler appliance, if the memory usage threshold is set to 75 percent instead of using the total available memory, it causes new TCP connections to bypass TCP optimization.

Unnecessary retransmissions due to missing SACK blocks

In a non-endpoint mode, when you send DUPACKS, if SACK blocks are missing for few out of order packets, triggers additional retransmissions from the server.

SNMP for number of connections bypassed optimization because of overload

The following SNMP ids have been added to a NetScaler appliance to track number of connections bypassed TCP optimization due to overload.

1. 1.3.6.1.4.1.5951.4.1.1.46.131 (tcpOptimizationEnabled). To track the total number of connections enabled with TCP optimization.
2. 1.3.6.1.4.1.5951.4.1.1.46.132 (tcpOptimizationBypassed). To track the total number of connections bypassed TCP Optimization.

Dynamic Receive Buffer

To maximize TCP performance, a NetScaler appliance can now dynamically adjust the TCP receive buffer size.

Reference Material

September 21, 2020

Use this reference information to get an in-depth understanding of the following NetScaler components:

NetScaler SNMP OIDs - Details of the SNMP OIDs that can be used to obtain information from a NetScaler appliance.
NetScaler Syslog Messages - Details of the Syslog messages given by the NetScaler appliance.

NetScaler CLI Commands - Details of the commands that can be used to configure the NetScaler appliance through the CLI. You can also view the details of each command in the NetScaler CLI, by entering the "man <ns-command-name>" command.

API Reference- Details of all operations that can be performed on the NetScaler appliance by using the REST API.

NetScaler Advanced Policy Expressions- Details of the expressions that can be used to define advanced policies.

Locations
Corporate Headquarters | 851 Cypress Creek Road Fort Lauderdale, FL 33309, United States
Silicon Valley | 4988 Great America Parkway Santa Clara, CA 95054, United States

© 2021 Citrix Systems, Inc. All rights reserved. Citrix, the Citrix logo, and other marks appearing herein are property of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered with the U.S. Patent and Trademark Office and in other countries. All other marks are the property of their respective owner(s).