

Traffic Management

Oct 13, 2015

Traffic Management

The following topics cover configuration and installation information for NetScaler traffic management features.

Cache Redirection	Analyzes incoming requests and forwards the requests for already cached data to cache servers. Dynamic HTTP requests and non-cacheable requests are forwarded to the origin servers. Cache redirection is a policy-based feature.
Content Switching	Analyzes client requests and redirects the requests to specific servers on the basis of geographical area, authorization credentials, and device from which the request was initiated.
DataStream	Ensures optimal distribution of traffic from the application and web servers to the database servers. Enables you to segment traffic according to information in the SQL query and on the basis of database names, user names, character sets, and packet size.
Domain Name System	Provides authoritative domain name server (ADNS server) functionality for a domain. The NetScaler appliance functions as a DNS end resolver and forwarder, and also helps in name resolution when fully qualified domain names are not configured.
Firewall Load Balancing	Distributes the traffic across multiple firewalls, providing fault tolerance, increased throughput, and high availability.
Global Server Load Balancing	Enables disaster recovery and ensures continuous availability of applications by protecting against points of failure in a wide area network (WAN).
Link Load Balancing	Load balances outbound traffic across multiple Internet connections to transmit packets seamlessly over the best possible link.
Load Balancing	Distributes user requests for web pages and other protected applications across multiple servers to prevent server overloading and failure. Load balancing also provides fault tolerance.
SSL Offload and Acceleration	Offloads SSL processing from a server to the NetScaler appliance to accelerate SSL transactions.
Web 2.0 Push	Offloads connection management from Web 2.0 servers. Instead of having to maintain a long-lived connection to each client, a Web 2.0 server can maintain a single connection to the NetScaler appliance. The appliance relays the server's data to waiting clients over the connections that it maintains with them.

Cache Redirection

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 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 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 resides directly in front of the clients. In a server-side deployment, the NetScaler 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

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 NetScaler 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

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 the simpler of two NetScaler expressions languages, called *classic expressions*. For a complete description of classic expressions and how to configure them, see

The NetScaler provides the following built-in 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.
bypass-dynamic-url	<p>Bypass the cache if the URL suggests that the content is dynamic, as indicated by the presence of any of the following extensions:</p> <ul style="list-style-type: none">o cgio aspo exeo cfmo exo shtmlo htx <p>Also bypass the cache if the URL starts with any of the following:</p> <ul style="list-style-type: none">o /cgi-bin/o /bin/o /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 .gif or .jpg.

Displaying the Built-in Cache Redirection Policies

Updated: 2013-08-23

You can display the available cache redirection policies by using the command line interface or the configuration utility.

To display the built-in cache redirection policies by using the command line interface

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_HEADE
3)      Cache-By-Pass RULE: (NS_EXT_CGI || NS_EXT_ASP || NS_EXT_EXE || NS_EXT_CFM || NS_EXT
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) Pol
Done
>

```

To display the built-in cache redirection policies by using the configuration utility

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.

Configuring a Cache Redirection Policy

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 uses the *classic policy* format. Each policy has a name and includes an expression or a set of expressions that are combined by using logical operators.

To add a cache redirection policy by using the command line interface

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>]

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
>
```

Policy with a compound expression:

```
> add cr policy Policy-CRD-2 -rule "REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == /*.cgi || RE
Done
> show cr policy Policy-CRD-2
      Cache-By-Pass RULE: REQ.HTTP.METHOD == POST && (REQ.HTTP.URL == '/*.cgi' || REQ.HTT
Done
>
```

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
>
```

To modify or remove a cache redirection policy by using the command line interface

- 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 ["Unbinding a Policy from a Cache Redirection Virtual Server"](#).

To configure a cache redirection policy with a simple expression by using the configuration utility

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:

- o Expression Type-General
- o Flow Type -REQ
- o Protocol -HTTP
- o Qualifier -URL
- o Operator - !=
- o Value* - /*.jpeg

The simple expression in the following example checks for an If-Modified-Since header in a request:

- o Expression Type -General
- o Flow Type -REQ
- o Protocol -HTTP
- o Qualifier -HEADER
- o Operator -EXISTS
- o Header Name -If-Modified-Since

5. When you are finished entering the expression, click OK or Create, and then click Close.

To configure a cache redirection policy with a compound expression by using the configuration utility

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:
 - o **Match Any Expression.** The policy matches the traffic if one or more individual expressions match the traffic.
 - o **Match All Expressions.** The policy matches the traffic only if every individual expression matches the traffic.
 - o **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 -) .)
- o **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 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.

- 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.

- 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.

- 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.

- 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.
Â	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.
Â	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.
	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.
Â	DESTPORT	Destination port of the TCP connection.
	MSS	Maximum segment size (MSS) of the TCP connection.
IP	SOURCEIP	Source IP address of the connection.
Â	DESTIP	Destination IP address of the connection.

- 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

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.

Configuring Transparent Redirection

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 response does match a policy, the appliance forwards the request, unchanged, to the web server.

For details on how to modify this default behavior, see ["Directing 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 NetScaler 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.

Enabling Cache Redirection and Load Balancing

The NetScaler cache redirection and load balancing features are not enabled by default. They must be enabled before any cache redirection configuration can take effect.

To enable cache redirection and load balancing by using the command line interface

At the command prompt, type the following command to enable cache redirection and load balancing and verify the settings:

- o enable ns feature cr lb
- o show ns feature

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)	NetScaler Push	push	OF

```
Done
>
```

To enable cache redirection and load balancing by using the configuration utility

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.

Configuring Edge Mode

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 off collection of statistics for the dynamically learned services and is typically used in transparent deployments for cache redirection.

To enable edge mode by using the command line interface

At the command prompt, type the following commands to enable edge mode and verify the setting:

- o enable ns mode Edge
- o show ns mode

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
>
```

To enable edge mode by using the configuration utility

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.

Configuring a Cache Redirection Virtual Server

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 NetScaler 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.

To add a cache redirection virtual server in transparent mode by using the command line interface

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>]`

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
>
```

To modify or remove a cache redirection virtual server by using the command line interface

- 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.

To add a cache redirection virtual server in transparent mode by using the configuration utility

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*â€™name
- Port*â€™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.

Binding Policies to the Cache Redirection Virtual Server

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.

To bind policies to a cache redirection virtual server by using the command line interface

At the command prompt, type:

- o bind cr vserver <name> -policyName <string>
- o show cr vserver [<name>]

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
>
```

To bind a user-defined policy to a cache redirection virtual server by using the configuration utility

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.

Unbinding a Policy from a Cache Redirection Virtual Server

When you unbind a policy from the cache redirection virtual server, the NetScaler appliance no longer applies the policy when evaluating client requests.

To unbind a policy from a cache redirection virtual server by using the command line interface

At the command prompt, type:

- o `unbind cr vserver <name> -policyName <string>`
- o `show cr vserver [<name>]`

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
>
```

To unbind a user-defined policy from a cache redirection virtual server by using the configuration utility

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.

Creating a Load Balancing Virtual Server

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."

To create a load balancing virtual server by using the command line interface

At the command prompt, type the following commands to create a load balancing virtual server and verify the configuration:

- o add lb vserver <name> <serviceType> [<IPAddress>] [<port>]
- o show lb vserver [<name>]

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
>
```

To create a load balancing virtual server 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:

- o Name*-name
- o IP Address*- IPAddress
- o 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 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.

Configuring an HTTP Service

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.

To configure an HTTP service by using the command line interface

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

- o add service <name> <IP> <serviceType> <port> -cacheType <cacheType>
- o show service [<name>]

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
>
```

To modify or remove a service by using the command line interface

- o 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.
- o To remove a service, use the rm service command, which accepts only the <name> argument.

To add an HTTP service 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:

- o Service Name*â€™name
- o Server*â€™ IP
- o Port*â€™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.

Binding/Unbinding a Service to/from a Load Balancing Virtual Server

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.

To bind a service to a load balancing virtual server by using the command line interface

At the command prompt, type:

- o bind lb vserver <name> <serviceName>
- o show lb vserver [<name>]

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
>
```

To unbind a service from a load balancing virtual server by using the command line interface

To unbind a service, use the unbind lb vserver command instead of bind lb vserver.

To bind/unbind a service from a load balancing virtual server by using the configuration utility

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.

Disabling the Use the Proxy Port Setting for Transparent Caching

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 NetScaler-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."](#)

Assigning a Port Range to the NetScaler

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.

To assign a source port range to a NetScaler appliance by using the command line interface

At the command prompt, type:

```
set ns param -crPortRange <startPortNumber-endPortNumber>
```

To assign a source port range to a NetScaler appliance by using the NetScaler configuration utility

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 NetScaler by typing a port number for Start Port and a port number for End Port.
4. Click OK.

Enabling Load Balancing Virtual Servers to Redirect Requests to Cache

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.

To enable load balancing virtual servers to redirect requests to the cache by using the command line interface

At the command prompt, type:

- o `set lb vserver <name> [-cacheable (YES | NO)]`
- o `show lb vserver [<name>]`

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
```

For transparent cache redirection, the NetScaler 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.

To turn off caching for a load balancing virtual server by using the command line interface

To turn off caching for a load balancing virtual, use the `unset lb vserver` command instead of `set lb vserver`. Specify a value or NO value for the `-cacheable` parameter.

To enable or disable load balancing virtual servers to redirect requests to the cache by using the configuration utility

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.

Configuring Forward Proxy Redirection

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 NetScaler 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 NetScaler compares the request with a policy for caching. If the request is not cacheable, the NetScaler 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 NetScaler forwards the request to a load balancing virtual server for the cache.

The NetScaler 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 NetScaler. 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 NetScaler, see ["Enabling Cache Redirection and Load Balancing."](#)

For details on how to create a load balancing virtual server, see ["Creating a Load Balancing Virtual Server."](#)

For details on how to configure services that represent the cache server, see ["Configuring an HTTP Service."](#)

For details on how to bind the service to a virtual server, see ["."](#)

For details on how to create a forward proxy cache redirection server, see ["Configuring 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 ["Configuring a Cache Redirection Policy."](#)

Creating a DNS Service

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.

To create a DNS service by using the command line interface

At the command line, type the following commands to create a DNS service and verify the configuration :

- o add service <name> <IP> <serviceType> <port>
- o show service [<name>]

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

To add an DNS service 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:

- o Service Name*â€™name
- o Server*â€™IP
- o Port*â€™port

* A required parameter

4. In the Protocol* drop down list, select a supported protocol (for example, **DNS**).
5. Click Create, and then click Close.

Creating a DNS Load Balancing Virtual Server

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.

To create a DNS load balancing virtual server by using the command line interface

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>]

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
>
```

To create a DNS load balancing virtual server 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, 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.

Binding the DNS Service to the Virtual Server

For the DNS server to respond to DNS requests, the service representing the DNS server must be bound to the DNS virtual server.

To bind the DNS service to the load balancing virtual server:

At the command prompt, type the following commands to bind the DNS service to the load balancing virtual server and verify the configuration:

- o bind lb vserver <name> <serviceName>
- o show lb vserver <name>

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
>
```

To unbind a DNS service from the load balancing virtual server:

Use the unbind lb vserver command instead of bind lb vserver.

To Bind/Unbind a DNS service to/from a load balancing virtual server by using the configuration utility

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.

Configuring a Client Web Browser to Use a Forward Proxy

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.

Configuring Reverse Proxy Redirection

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 NetScaler 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 NetScaler, see ["Enabling Cache Redirection and Load Balancing."](#)

For details on how to create a load balancing virtual server, see ["Creating a Load Balancing Virtual Server."](#)

For details on how to configure services that represent the cache server, see ["Configuring an HTTP Service."](#)

For details on how to bind the service to a virtual server, see ["."](#)

For details on how to create a reverse proxy cache redirection server, see ["Configuring 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 ["Binding Policies to the Cache Redirection Virtual Server."](#)

Configuring 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/sports/index.html](#) 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.

To configure a mapping policy for reverse proxy mode by using the command line interface

At the command prompt, type the following command to add a policy map and verify the configuration:

- o add policy map <mapPolicyName> -sd <string> [-su <string>] [-td <string>] [-tu <string>]
- o show policy map [<mapPolicyName>]

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
>
```

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 /n
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
>
```

To configure a mapping policy for reverse proxy mode by using the configuration utility

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:

- o Name*- mapPolicyName
- o Source Domain*-sd
- o Target Domain*-td
- o Source URL-su
- o Target URL-tu

* A required parameter

4. Click Create, and then click Close. The Map pane displays the new mapping policy.

To bind the mapping policy to the cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to bind the mapping policy to the cache redirection virtual server and verify the configuration:

- o bind cr vserver <name> -policyName <string> [<targetVserver>]
- o show cr vserver <name>

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
>
```

To bind the mapping policy to the cache redirection virtual server by using the configuration utility

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

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 non-cacheable 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 NetScaler 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 NetScaler 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 NetScaler sends all cacheable HTTP traffic to a transparent cache farm. Clients access the Internet through the NetScaler, which is configured as a Layer 4 switch that receives traffic on port 80.

The NetScaler can direct requests for images (for example, .gif 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 NetScaler, see "[Configuring Cache Redirection](#)".

Enabling Content Switching

To configure selective cache redirection, after you enable both the load balancing and cache redirection features on the NetScaler, you must enable content switching.

To enable content switching by using the command line interface

At the command prompt, type:

- enable ns feature CS
- show ns feature

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)	NetScaler Push	push	OFF

```
Done
```

To enable cache redirection and load balancing by using the configuration utility

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.

Configuring a Load Balancing Virtual Server for the Cache

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 ["Creating a Virtual Server"](#).

For details on how to configure services that represent the cache server, see ["Configuring an HTTP Service."](#)

For details on how to bind the service to a virtual server, see ["Binding/Unbinding a Service to/from a Load Balancing Virtual Server."](#)

For details on how to create a transparent proxy cache redirection server, see ["Configuring 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 ["Binding Policies to the Cache Redirection Virtual Server."](#)

Configuring a Cache Redirection Policy for a Specific Type of Content

To identify requests that contain a .gif 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 ".gif" or ".jpeg" extensions.

To configure cache redirection for a specific type of content, configure a policy that uses a simple expression, as described in ["Configuring a Cache Redirection Policy."](#)

Configuring Policies for Content Switching

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 .gif 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 ["](#).

You must first create the content switching policy and then bind it to the cache redirection virtual server.

To create a content switching policy by using the command line interface

At the command line, type:

- add cs policy <policyName> [-url <string> | -rule <expression>]
- show cs policy [<policyName>]

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 == '/ * .gif'"
Done
> show cs policy Policy-CS-GIF
      Rule: REQ.HTTP.URL == '/ * .gif'          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 /*.gif
Done
> show cs policy Policy-CS-GIF-URL
      URL: /*.gif          Policy: Policy-CS-GIF-URL
      Hits: 0
Done
>
```

To create a URL-based content switching policy by using the configuration utility

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.

To create a rule-based content switching policy by using the configuration utility

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.
 - o If you want to use default syntax, accept the default and proceed to the next step.
 - o 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.
 - o If you are using classic syntax and need further instructions, see "[Configuring Classic Policies and Expressions](#)."
 - o If you are using the default syntax and need further instructions, see "[Configuring Default Syntax Expressions: Getting Started](#)."
7. Click Create and click Close. The policy you created appears in the Content Switching Policies pane.

To bind the content switching policy to a cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to bind the content switching policy to a cache redirection virtual server and verify the configuration:

- o bind cs vserver <name> <targetVserver> [-policyName <string>]
- o show cs vserver [<name>]

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
>
```

To bind the content switching policy to a cache redirection virtual server by using the configuration utility

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.

Configuring Precedence for Policy Evaluation

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

To configure precedence for policy evaluation by using the command line interface

At the command prompt, type the following commands to configure precedence for policy evaluation and verify the configuration:

- o set cr vsrver <name> [-precedence (RULE | URL)]
- o show cr vsrver <name>

Example

```
> set cr vsrver Vserver-CRD-1 -precedence URL
Done
> show cr vsrver 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
>
```

To configure precedence for policy evaluation by using the configuration utility

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.

Administering a Cache Redirection Virtual Server

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.

Viewing Cache Redirection Virtual Server Statistics

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

To view statistics for a cache redirection virtual server by using the command line interface

At the command prompt, type:

stat cr vsrver [<name>]

Example

```
> stat cr vsrver 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

>

To view statistics for a cache redirection virtual server by using the configuration utility

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

To 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.

Enabling or Disabling a Cache Redirection Virtual Server

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.

To Enable or Disable a cache redirection virtual servers by using the command line interface

At the command line, type one of the following commands:

- o enable cr vserver <name>
- o show cr vserver <name>
- o disable cr vserver <name>
- o show cr vserver <name>

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
>
```

To Enable or Disable a cache redirection virtual servers by using the configuration utility

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.

Directing Policy Hits to the Cache Instead of the Origin

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 `cachedirection` type as POLICY.

Possible values: CACHE, ORIGIN

Default value: ORIGIN

To change the destination for a policy hit to the origin or the cache by using the command line interface

At the command prompt, type the following commands to change the destination for a policy hit and verify the configuration:

- o `set cr vsrver <name> [-onPolicyMatch (ORIGIN | CACHE)]`
- o `show cr vsrver <name>`

Example

```
> set cr vsrver Vserver-CRD-1 -onPolicyMatch CACHE
Done
> show cr vsrver 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
```

To change the destination for a policy hit to the origin or the cache by using the configuration utility

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.

Backing Up a Cache Redirection Virtual Server

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

To specify a backup cache redirection virtual server by using the command line interface

At the command prompt, type the following commands to specify a backup cache redirection virtual server and verify the configuration:

- o set cr vserver <name> [-backupVServer <string>]
- o show cr vserver <name>

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
```

To specify a backup cache redirection virtual server by using the configuration utility

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.

Managing Client Connections for a Virtual Server

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 NetScaler 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, NetScaler always responds.
- When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.
- When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

This document includes the following information:

- [Configuring Client Timeout](#)
- [Inserting Via Headers in the Requests](#)
- [Reusing TCP Connections](#)
- [Configuring Delayed Connection Cleanup](#)

Configuring Client Timeout

Updated: 2013-08-22

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.

To configure client timeout by using the command line interface

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>`

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
```

To configure client timeout by using the configuration utility

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.

Inserting Via Headers in the Requests

Updated: 2013-08-23

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

To enable or disable Via-header insertion in client requests by using the command line interface

At the command prompt, type:

- o set cr vserver <name> [-via (ON|OFF)]
- o show cr vserver <name>

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
>
```

To enable or disable Via-header insertion in client requests by using the configuration utility

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.

Reusing TCP Connections

Updated: 2013-11-08

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 NetScaler. 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

To enable or disable the reuse of TCP connections by using the command line interface

At the command prompt, type:

- o set cr vsrver <name> [-reuse (ON|OFF)]
- o show cr vsrver <name>

Example

```
> set cr vsrver Vserver-CRD-1 -reuse ON
Done
> show cr vsrver 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
```

To enable or disable the reuse of TCP connections by using the configuration utility

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.

Configuring Delayed Connection Cleanup

Updated: 2013-08-22

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

To enable of disable the down state flush option by using the command line interface

At the command prompt, type the following commands to configure delayed connection clean up and verify the configuration:

- o set cr vsrver <name> [-downStateFlush (ENABLED | DISABLED)]
- o show cr vsrver <name>

Example

```
> set cr vsrver Vserver-CRD-1 -downStateFlush ENABLED
Done
```

```

> show cr vsriver 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

```

To enable or disable the reuse of TCP connections by using the configuration utility

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

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, NetScaler 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 NetScaler from which it received the request and send the response to the same NetScaler.

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 NetScaler 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 NetScaler, 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 NetScaler 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 NetScaler and bind all the services to the virtual server.

For each lower-tier NetScaler appliance,

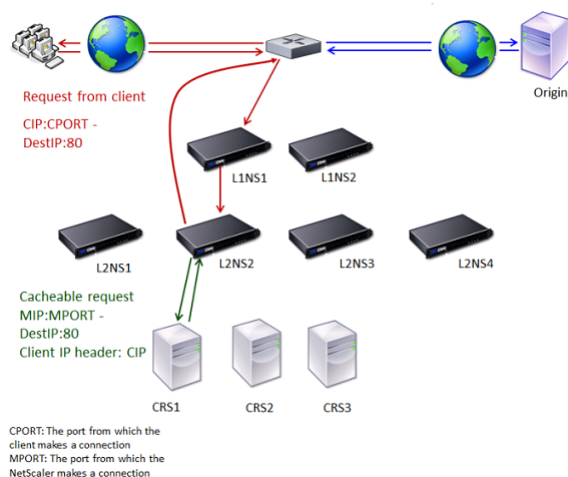
- Configure the cache redirection port range on the NetScaler. Assign an exclusive range to each lower-tier NetScaler.
- Configure a load balancing virtual server and enable MAC-based redirection.

- Create a service for each cache server that is to be load balanced by this NetScaler. 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



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.

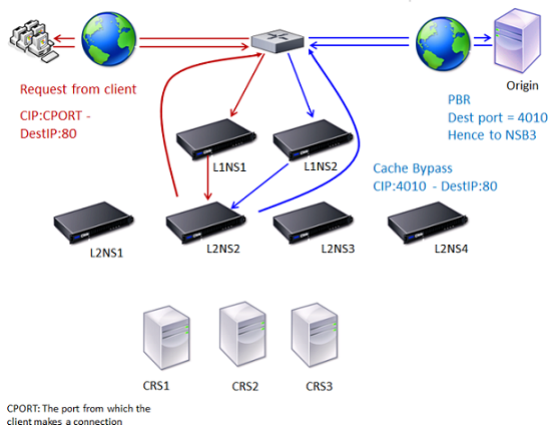
Traffic Flow

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 NetScaler 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



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.

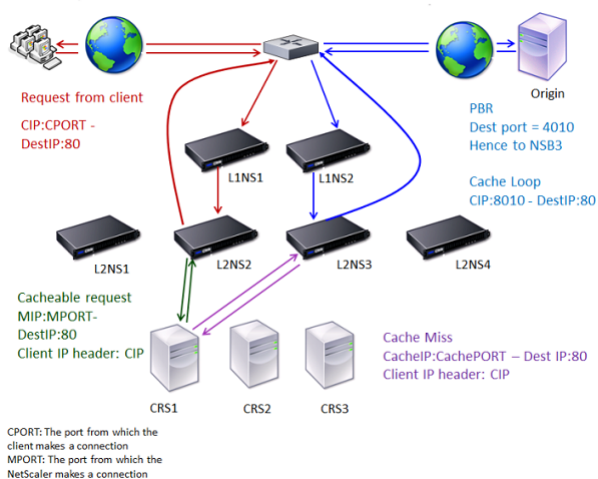
Traffic Flow

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 NetScaler, L1NS2.
5. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler 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 NetScaler 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



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 NetScaler 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 NetScaler, L1NS2.
7. According to the PBR policies, L1NS2 forwards the traffic to the appropriate NetScaler in the lower tier, L2NS3.
8. L2NS3 forwards the response to the router.
9. The router forwards the response to Client A.

Configuring the Upper-Tier NetScaler Appliances

Configure each of the upper-tier NetScaler appliances as follows.

To configure an upper-tier appliance for n-tier cache redirection by using the command line interface

At the command prompt, type the following commands:

- o add service <name>@ <serviceIP> <serviceType> <port>

Run this command for each service to be added.

- o add lb vserver <name>@ ANY * <port> -persistenceType <persistenceMethod> -lbMethod <lbMethod> -m MAC -sessionless ENABLED -cltTimeout <client_Timeout_Value>
- o bind lb vserver <name>@ <serviceName>

Run this command for each service to be bound.

- o enable ns mode l3
- o add ns pbr <name> <action> -srcPort <sourcePortNumber> -destPort <startPortNumber-endPortNumber> -nextHop <serviceIPAddress> -protocol TCP
- o apply ns pbrs

Run this command after adding all the necessary PBRs.

To configure an upper-tier appliance for n-tier cache redirection by using the configuration utility

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.
 - a. In the Policy-Based Routing (PBRs) pane, click Add.
 - b. Type a name for the PBR.
 - c. Select the action as Allow.
 - d. In the Next Hop box, type the IP address of the service, which represents a lower-tier NetScaler.
 - e. Select TCP from the Protocol drop-down list.
 - f. Type the source port and the range of the destination port corresponding to the lower-tier NetScaler being added.
 - g. Click Create.
 - h. In the details pane, select the PBR and click Apply.
 - i. Repeat Step (i) to Step (vii) for each lower-tier NetScaler.
3. Create a service for each lower-tier NetScaler:
 - a. Navigate to Traffic Management > Load Balancing > Services.
 - a. In the details pane, click Add.
 - b. Specify the name, protocol, IP address, and port. The protocol should be ANY.
 - c. Click Create.
4. Configure a load balancing virtual server:
 - a. Navigate to Traffic Management > Load Balancing > Virtual Servers.
 - a. In the details pane, click Add.
 - b. Specify the name, protocol, IP address, and port. The protocol should be ANY and the IP address should be *.
 - c. In the Services tab, select the services that represent the lower-tier NetScaler appliances.
 - d. In the Advanced tab, select the Redirection Mode as MAC Based and select the Sessionless check box.
 - e. Click Create.

Configuring the Lower-Tier NetScaler Appliances

Configure each of the lower-tier NetScaler appliances as follows.

To configure a lower-tier appliance for n-tier cache redirection by using the command line interface

At the command prompt, type the following commands:

- o add service <name>@ <cacheServiceIP> <serviceType> <port> -cip ENABLED "ClientIP" -cachetype transparent

Repeat for each cache server.

- o add lb vserver <name>@ <serviceType> -m MAC
- o bind lb vserver <name>@ <cacheServiceName>

Repeat for each cache server.

- o add cr vserver <name> <serviceType> * <port> -srcIPExpr "HTTP.REQ.HEADER(\"ClientIP\")" -originusip ON -usePortRange ON
- o set ns param-crPortRange <startPortNumber-endPortNumber>

To configure a lower-tier appliance for n-tier cache redirection by using the configuration utility

1. Create a service for each cache server. To create a service:
 - a. Navigate to Traffic Management > Load Balancing > Services.
 - a. In the details pane, click Add, and specify the name and protocol. Clear the Directly Addressable check box.
 - b. In the Advanced tab, select the Override Global check box and the Client IP check box, and then in the Header box, type ClientIP.
 - c. In the Cache Type box, select Transparent Cache.
 - d. Click Create.
2. Configure a load balancing virtual server:
 - a. Navigate to Traffic Management > Load Balancing > Virtual Services.
 - a. In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be an asterisk (*).
 - b. In the Services tab, select the services that represent the cache servers.
 - c. In the Advanced tab, for Redirection Mode, select MAC Based.
 - d. Click Create.
3. Configure a cache redirection virtual server:
 - a. Navigate to Traffic Management > Load Balancing > Virtual Services.
 - a. In the details pane, click Add and specify the name, protocol, IP address, and port. The IP address should be *
 - b. For Cache Type, select Transparent.
 - c. 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").
 - d. Click Create.
4. Assign a source port range for the NetScaler:
 - 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 NetScaler by typing a port number for Start Port and a port number for End Port.
 - d. Click OK.

Content Switching

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 Citrix NetScaler content switching feature enables the NetScaler 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

Updated: 2013-08-22

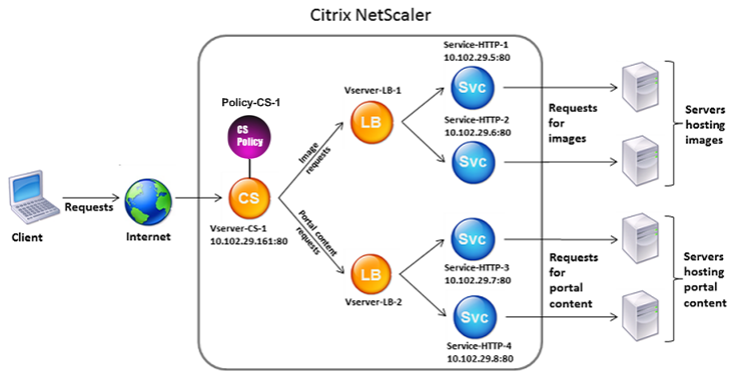
Content Switching enables the NetScaler appliance to direct requests sent to the same Web host to different servers with different content. For example, you can configure the appliance to direct requests for dynamic content (such as URLs with a suffix of .asp, .dll, or .exe) to one server and requests 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



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

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 re-enabling 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

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
```

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

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
```

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

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 "[Setting Up Basic Load Balancing](#)."

Configuring a Content Switching Action

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 switching 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

Updated: 2013-08-22

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:

- o add cs action <name> -targetLBVserver <string> [-comment <string>]
- o show cs action <name>

Example

```
> add cs action mycsaction -targetLBVserver mylbvserver -comment "Forwards requests to mylbv
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
>
```

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>`

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
    æ|
Done
>
```

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

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 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 classic policies and expressions, see ["Configuring Classic Policies and Expressions."](#)
For information about Default Syntax policies, see ["Configuring Default Syntax Expressions."](#)

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 "/sports/*"  
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
```

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
```

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

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 command line or the configuration utility to configure content switching policy labels. In the NetScaler command-line interface (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 configuration utility, 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> <cspolicylabelType>
```

Example

```
add cs policylabel testpollab http
```

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
```

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
1)    Policy Name: test_Pol
    Priority: 100
    Target Virtual Server: LBVIP
```

Note: If a policy is configured with an action, the target virtual server (targetVserver), goto priority expression (gotoPriorityExpression), and invoke (invoke) parameters are not required. If a policy is not configured 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:

- o unbind cs policylabel <labelName> <policyName>
- o 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
```

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

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>)]
```

Example

```
bind cs vserver csw-vip2 -policyname csw-ape-policy2 -priority 14 -gotoPriorityExpression NE

bind cs vserver csw-vip3 -policyname rewrite-policy1 -priority 17 -gotoPriorityExpression
'q.header("a").count' -flowtype REQUEST -invoke policylabel label1

bind cs vserver Vserver-CS-1 Vserver-LB-1 -policyname Policy-CS-1 -priority 20
```

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

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 Policy-Based 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 cspoll -logAction csLogAction
Done
> show cs policy cspoll

Policy: cspoll      Rule: TRUE      Action: csact1
LogAction: csLogAction
Hits: 0

1)      CS Vserver: csvs1
Priority: 10
Done
>
```

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

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

Updated: 2013-10-31

You can view the properties of content switching virtual servers that you have configured on the NetScaler. 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:

- o show cs vserver
- o 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

1) Rewrite Policy Name: __ESNS_REWRITE_POLICY Priority: 2147483647
GotoPriority Expression: END
Flowtype: REQUEST

1) Cache Policy Name: dfbx Priority: 10
GotoPriority Expression: END
Flowtype: REQUEST

1) 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
>
```

```
2.
show cs vserver
1) Vserver-CS-1 (10.102.29.161:80) - HTTP Type: CONTENT
State: UP
â€|
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED

2) apubendpt (10.111.111.1:80) - HTTP Type: CONTENT
State: UP
â€|
Client Idle Timeout: 180 sec
Down state flush: DISABLED
â€|

3) apubendpt1 (10.111.111.2:80) - HTTP Type: CONTENT
State: UP
â€|
Disable Primary Vserver On Down : DISABLED
Appflow logging: DISABLED
Port Rewrite : DISABLED
State Update: DISABLED
â€|
```

Viewing Content Switching Policies

Updated: 2013-08-22

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
```

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

Updated: 2013-08-22

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.

- o 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:
 - o Click the Zoom In and Zoom Out icons to increase or decrease the viewable area.
 - o Click the Save Image icon to save the graph as an image file.
 - o 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:
 - o 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.
 - o 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.
 - o 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 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

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.

To customize the basic content switching configuration, see the following sections:

- [Configuring Case Sensitivity for Policy Evaluation](#)
- [Setting the Precedence for Policy Evaluation](#)
- [Support for Multiple Ports for HTTP and SSL Type Content Switching Virtual Servers](#)
- [Configuring per-VLAN Wildcarded Virtual Servers](#)
- [Configuring the Microsoft SQL Server Version Setting](#)

Configuring Case Sensitivity for Policy Evaluation

Updated: 2013-10-31

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
```

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

Updated: 2013-10-31

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
```

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

Updated: 2014-05-21

You can configure the NetScaler ADC 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
RHlstate: PASSIVE
TD: 0

```

Done

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

Updated: 2013-10-31

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
```

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

Updated: 2013-08-22

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:

- o set cs vserver <name> -mssqlServerVersion <mssqlServerVersion>
- o 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
>
```

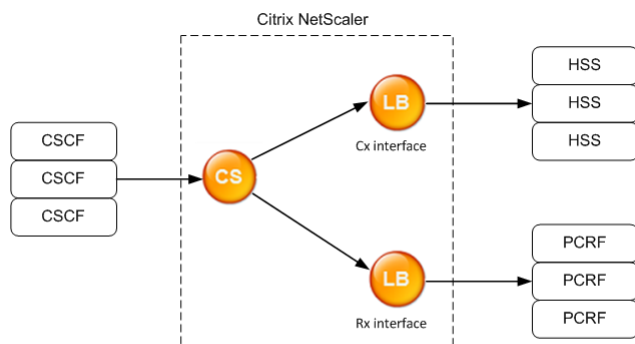
To set the Microsoft SQL Server version parameter by using the configuration utility

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

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 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]
```

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
```

Protecting the Content Switching Setup against Failure

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:

- **Configure a backup content switching virtual server**
- **Configure spillover for preventing the overloading of the primary and diverting excess traffic to the backup virtual server**
- **Specify a redirect URL, the URL to which the content is switched if both the primary and backup content switching virtual servers are DOWN**
- **Enable the State Update option for marking a content switching virtual server as DOWN when the load balancing virtual server is DOWN**
- **Flush the surge queues when the queues become too long**

Configuring a Backup Virtual Server

Updated: 2013-11-08

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)
```

Example

```
set cs vserver Vserver-CS-1 -backupVserver Vserver-CS-2 -disablePrimaryOnDown ON
```

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

Updated: 2013-11-04

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>
```

Example

```
set cs vserver Vserver-CS-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -s
```

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

Updated: 2015-03-19

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>
```

Example

```
set cs vserver Vserver-CS-1 -redirectURL http://www.newdomain.com/mysite/maintenance
```

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

Updated: 2014-12-12

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
```

Example

```
add cs vserver csw_vserver HTTP 10.18.250.154 80 -stateupdate ENABLED -cltTimeout 180
```

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

Updated: 2013-12-04

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 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>]
```

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 SVC
2.

```
flush ns surgeQ
```

The above command flushes all the surge queues on the appliance.

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

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.

To manage a content switching setup, see the following sections:

- [Unbinding Policies from the Content Switching Virtual Server](#)
- [Removing Content Switching Virtual Servers](#)
- [Disabling and Re-Enabling Content Switching Virtual Servers](#)
- [Renaming Content Switching Virtual Servers](#)
- [Managing Content Switching Policies](#)
- [Modifying a Content Switching Configuration by Using the Visualizer](#)

Unbinding Policies from the Content Switching Virtual Server

Updated: 2014-09-03

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
```

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

Updated: 2013-10-31

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
```

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

Updated: 2013-10-31

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
```

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

Updated: 2013-10-31

You can rename a content switching virtual server without unbinding it. The new name is propagated automatically to all affected parts of the NetScaler 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
```

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

Updated: 2013-08-22

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.

Table 1. Examples of URL-Based Policies

Type of URL-Based Policy	Specifies
Domain and Exact URL	<p>Requests must match the configured domain name and configured URL (an exact prefix match if only the prefix is configured; or an exact match of the prefix and suffix if both the prefix and suffix are configured).</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -url /sports/tennis/index.html -domain "www.domainxyz.com"</pre>
	<p>Requests must match the exact domain name and a partial prefix of the configured URL.</p>

Domain and Wild Card URL	<p>Example:</p> <pre>add cs policy Policy-CS-1 -url /*.jsp -domain "www.domainxyz.com"</pre>
Domain Only	<p>Requests need match only the configured domain name.</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -domain "www.domainxyz.com"</pre>
The Exact URL	<p>The incoming URL must exactly match the URL specified by the policy. If only a URL prefix rule is configured, there must be an exact prefix match with the incoming URL. If a URL prefix and suffix-based rule is configured, there should be an exact match of the prefix and suffix with the incoming URL.</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -url /sports/tennis/index.html</pre>
Prefix Only (Wild Card URL)	<p>All the incoming URLs must start with the configured prefix.</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -url /sports*</pre> <p>sports/*â€• matches all URLs under /sports â€œ/sports*â€• matches all URLs whose prefix match â€œ/sportsâ€• starting from the beginning of a URL</p>
Suffix Only (Wild Card URL)	<p>All incoming URLs must end with the configured URL suffix.</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -url /*.jsp</pre> <p>â€œ/*â€• matches all URLs whose file extension is â€œ.jspâ€•</p>
Prefix and Suffix (Wild Card URL)	<p>All incoming URLs must start with the configured prefix and end with the configured suffix.</p> <p>Example:</p> <pre>add cs policy Policy-CS-1 -url /sports/*.jsp</pre>

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
```

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.

Modifying a Content Switching Configuration by Using the Visualizer

Updated: 2013-08-22

You can use the Visualizer to modify a load balancing virtual server to which the content switching virtual server is bound. You can also modify a service or group of similar services, or a monitor. For more information, see "[The Load Balancing Visualizer](#)."

Managing Client Connections

To ensure efficient management of client connections, you can configure the content switching virtual servers on the NetScaler appliance to use the following features:

- **Redirecting client requests to a cache**
- **Enabling delayed cleanup of virtual server connections**
- **Rewriting ports and protocols for redirection**
- **Inserting the IP address and port of a virtual server in the request header**
- **Setting a time-out value for idle client connections**
- **Identifying Connections with the 4-tuple and Layer 2 Connection Parameters**
- **Configuring the ICMP Response.** You can configure the NetScaler 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, NetScaler always responds.

When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.

When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

Redirecting Client Requests to a Cache

Updated: 2013-10-31

The NetScaler 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
```

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

Updated: 2013-10-31

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
```


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

Updated: 2013-10-31

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
```

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

Updated: 2013-10-31

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
```

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

Updated: 2013-10-31

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
```

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

Updated: 2013-08-22

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:

- o set cs vserver <name> -l2Conn (ON | OFF)
- o 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
>
```

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

If the content switching 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 Content Switching

Updated: 2013-07-22

For best results, use the following resources to troubleshoot a content switching issue on a NetScaler appliance:

- Configuration file
- Relevant newslog 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 trace files
- An SSH utility for command line access
- A HyperTerminal utility to access the console

Troubleshooting Content Switching Issues

Updated: 2013-08-02

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.

Domain Name System

You can configure the Citrix 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 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, 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.

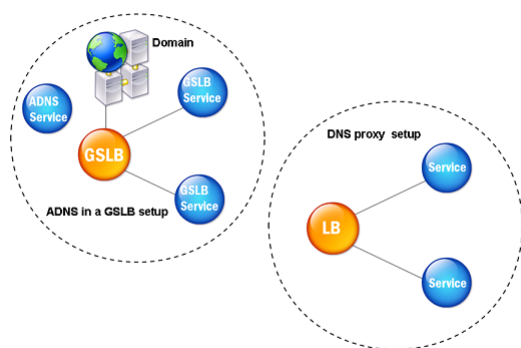
The NetScaler can be configured as the authority for a domain. To do this, you add valid SOA and NS records for the domain.

An ADNS server is a DNS server that contains complete information about a zone.

To configure the NetScaler 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, the NetScaler 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 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:

- o 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.
- o 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 dn
```

The domain, abc.com is linked to an NS record as follows:

```
add dns nsrec abc.com. ns1
```

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
```

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
```

In the third DNS response, the third IP address, 1.1.1.3 is served as the first record:

```
ns1. 1H IN A 1.1.1.3 ns1. 1H IN A 1
```

Configuring DNS Resource Records

You configure resource records on the Citrix® NetScaler® 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 and the number of records (per record type) that you can configure for a domain on the NetScaler.

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

Creating SRV Records for a Service

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.

To add an SRV record by using the command line interface

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>`

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
>
```

To modify or remove an SRV record by using the command line interface

- 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.

To configure an SRV record by using the configuration utility

Navigate to Traffic Management > DNS > Records > SRV Records and create an SRV record.

Creating AAAA Records for a Domain Name

An AAAA resource record stores a single IPv6 address.

To add an AAAA record by using the command line interface

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>`

Example

```
> add dns aaaaRec www.example.com 2001:0db8:0000:0000:0000:0000:1428:57ab
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
>
```

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.

To add an AAAA record by using the configuration utility

Navigate to Traffic Management > DNS > Records > AAAA Records and create an AAAA record.

Creating Address Records for a Domain Name

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.

To add an Address record by using the command line interface

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>`

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
>
```

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.

To add an Address record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Address Records and create an Address record.

Creating MX Records for a Mail Exchange Server

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.

To add an MX record by using the command line interface

At the command prompt, type the following commands to add an MX record and verify the configuration:

- o add dns mxRec <domain> -mx <string> -pref <positive_integer> [-TTL <secs>]
- o show dns mxRec <domain>

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
>
```

To modify or remove an MX record by using the command line interface

- o 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.
- o 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.
- o 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.

To add an MX record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Mail Exchange Records and create an MX record.

Creating NS Records for an Authoritative Server

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.

To create an NS record by using the command line interface

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>

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
>
```

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.

To create an NS record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Name Server Records and create an NS record.

Creating CNAME Records for a Subdomain

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.

To add a CNAME record by using the command line interface

At the command prompt, type the following commands to create a CNAME record and verify the configuration:

- o add dns cnameRec <aliasName> <canonicalName> [-TTL <secs>]
- o show dns cnameRec <aliasName>

Example

```
> add dns cnameRec www.example.com www.exampnenw.com
Done
> show dns cnameRec www.example.com
      Alias Name      Canonical Name  TTL
1)      www.example.com      www.exampnenw.com      5 secs
Done
>
```

To remove a CNAME record for a given domain, type the rm dns cnameRec command and the alias of the domain name.

To add a CNAME record by using the configuration utility

Navigate to Traffic Management > DNS > Records > Canonical Records and create a CNAME record.

Caching of CNAME Records

Updated: 2015-05-26

NetScaler ADC 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:

- o NetScaler should be deployed in a proxy mode
- o 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
- o The response from the back-end server cannot be a No-data or NX-Domain
- o The response from the back-end server has to be a authoritative response

Creating NAPTR Records for Telecommunications Domain

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.

NetScaler ADCs 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.

To create a NAPTR record by using command line interface

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>](regex<expressions>|-replacement<string>)[-TTL <secs>]
```

To remove a NAPTR record by using command line interface

```
rm dns naptrRec<domain> (<order> <preference> [-flags <string>] [-services <string>] (-regex <expression> | -replacement <string>)) | -recordId <positive_integer>@)
```

To configure a NAPTR record using configuration utility

Navigate to Traffic Management > DNS > Records > NAPTR Records and create an NAPTR record.

Creating PTR Records for IPv4 and IPv6 Addresses

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.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.1.2.3.4.IP6.ARPA.

To add a PTR record by using the command line interface

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>

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
>
```

To remove a PTR record, type the rm dns ptrRec command and the reverse domain name associated with the PTR record

To add a PTR record by using the configuration utility

Navigate to Traffic Management > DNS > Records > PTR Records and create a PTR record.

Creating SOA Records for Authoritative Information

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.

To create an SOA record by using the command line interface

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>

Example

```
> add dns soaRec example.com -originServer nameserver1.example.com -contact admin.example.co
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
>
```

To modify or remove an SOA record by using the command line interface

- 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.

To configure an SOA record by using the configuration utility

Navigate to Traffic Management > DNS > Records > SOA Records and create an SOA record.

Creating TXT Records for Holding Descriptive Text

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.

To create a TXT resource record by using the command line interface

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>]`

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
      "Contact: Mark"
      "Email: mark@example.com"
Done
```

To remove a TXT resource record by using the command line interface

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>]`

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
      "Contact: Evan"
      "Email: evan@example.com"
2)      Domain : www.example.com      Record id: 14373      TTL : 36000 secs
      "Contact: Mark"
      "Email: mark1@example.com"
Done
```

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
      "Contact: Mark"
      "Email: mark1@example.com"
Done
```

To configure a TXT record by using the configuration utility

Navigate to Traffic Management > DNS > Records > TXT Records and create a TXT record.

Viewing DNS Statistics

You can view the DNS statistics generated by the Citrix® NetScaler® appliance. The DNS statistics include runtime, configuration, and error statistics.

To view DNS records statistics by using the command line interface

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
>
```

To view DNS records statistics by using the configuration utility

1. Navigate to Traffic Management > DNS.
2. In the details pane, click Statistics.

Configuring a DNS Zone

A DNS zone entity on the Citrix® NetScaler® 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 [Offloading 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 [Configuring 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.

To create a DNS zone on the NetScaler appliance by using the command line interface

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>]

Example

```
> add dns zone example.com -proxyMode Yes
Done
> show dns zone example.com
    Zone Name : example.com
    Proxy Mode : YES
Done
>
```

To modify or remove a DNS zone by using the command line interface

- 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.

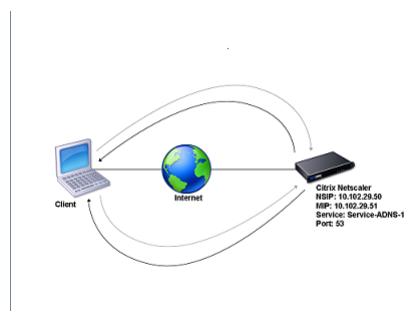
To configure a DNS zone by using the configuration utility

Navigate to Traffic Management > DNS > Zones and create a DNS zone.

Configuring the NetScaler as an ADNS Server

You can configure the Citrix® 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. Normally, the ADNS service uses the Mapped IP address (MIP). However, you can configure the ADNS service with any NetScaler-owned 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.

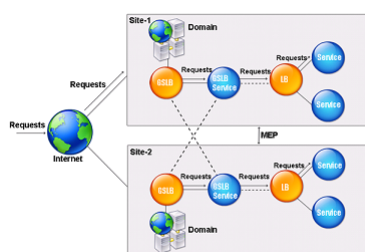
Table 1. Example of ADNS Service Configuration

Entity type	Name	IP address	Type	Port
ADNS Service	Service-ADNS-1	10.102.29.51	ADNS	53

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



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.

This document includes the following information:

- [Creating an ADNS Service](#)
- [Configuring the ADNS Setup to Use TCP](#)
- [Adding DNS Resource Records](#)
- [Removing ADNS Services](#)
- [Configuring Domain Delegation](#)

Creating an ADNS Service

Updated: 2014-11-14

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 [Configuring Services](#).

Note: You can configure the ADNS service to use MIP, 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.

Configuring the ADNS Setup to Use TCP

Updated: 2013-08-26

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 Transmission Control Protocol (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.

Adding DNS Resource Records

Updated: 2013-08-26

After you create an ADNS service, you can add DNS records. For instructions on adding DNS records, see [Configuring DNS Resource Records](#).

Removing ADNS Services

Updated: 2013-08-27

For instructions on removing services, see [Load Balancing](#).

Configuring 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.

Creating an SOA Record

For instructions on configuring SOA records, see [Creating SOA Records for Authoritative Information](#).

Creating an NS Record for a Name Server

For instructions on configuring an NS record, see [Creating NS Records for an Authoritative Server](#). In the Name Server drop-down list, select the primary authoritative name server, for example, ns1.abc.com.

Creating an Address Record

For instructions on configuring Address records, see [Creating 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.

Creating an NS Record for Domain Delegation

For instructions on configuring NS records, see [Creating 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.

Creating 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.

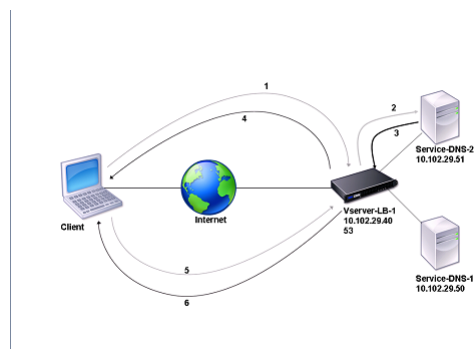
For instructions on adding glue records for a subdomain, see the procedure for adding an Address (A) record, [Configuring DNS Resource Records](#).

For instructions on configuring Address records, see [Creating Address Records for a Domain Name](#). 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.

Configuring the NetScaler as a DNS Proxy Server

As a DNS proxy server, the Citrix® NetScaler® 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 NetScaler 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.

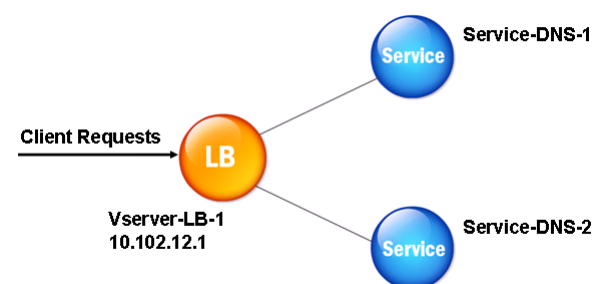
The following table lists sample names and the values of the entities that need to be configured on the NetScaler.

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
	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. For information about configuring load balancing services and virtual servers, see "Load Balancing", and then configure DNS proxy setup.

This document includes the following information:

- [Creating a Load Balancing Virtual Server](#)
- [Creating DNS Services](#)
- [Binding a Load Balancing Virtual Server to DNS Services](#)
- [Configuring the DNS Proxy Setup to Use TCP](#)
- [Enabling Caching of DNS Records](#)
- [Adding DNS Resource Records](#)
- [Removing a Load Balancing DNS Virtual Server](#)
- [Limiting the Number of Concurrent DNS Requests on a Client Connection](#)

Creating a Load Balancing Virtual Server

Updated: 2014-12-29

To configure a DNS Proxy on the NetScaler ADC, 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](#)".

Creating DNS Services

Updated: 2013-08-26

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](#)".

Binding a Load Balancing Virtual Server to DNS Services

Updated: 2013-09-13

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 "".

Configuring the DNS Proxy Setup to Use TCP

Updated: 2013-08-26

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 Citrix® NetScaler® 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.

Enabling Caching of DNS Records

Updated: 2013-08-27

To complete the process of configuring a DNS proxy on the NetScaler, you must enable caching of DNS records. You must also specify minimum and maximum time-to-live (TTL) values for the records that are cached. The TTL values are measured in seconds.

To enable caching of DNS records by using the command line interface

At the command prompt, type the following commands to enable caching of DNS records and verify the configuration:

- o set dns parameter -cacheRecords Yes
- o show dns parameter

Example

```
> set dns parameter -cacheRecords YES
Done
> show dns parameter
.
.
.
Cache Records : YES
.
.
.
Done
>
```

To enable caching of DNS records by using the configuration utility

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 records caching check box, and then click OK.

Configuring 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 2147483647. 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.

To specify the minimum and/or maximum TTL by using the command line interface

At the NetScaler command prompt, type the following commands to specify the minimum and maximum TTL and verify the configuration:

- o set dns parameter [-minTTL <secs>] [-maxTTL <secs>]
- o show dns parameter

Example

```
> set dns parameter -minTTL 1200 -maxTTL 1800
Done
> show dns parameter
DNS parameters:
DNS retries: 5
Minimum TTL: 1200           Maximum TTL: 1800
.
.
.
Done
>
```

To specify the minimum and/or maximum TTL by using the configuration utility

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.

Flushing 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.

To delete all proxy records by using the command line interface

At the NetScaler command prompt, type:

```
flush dns proxyRecords
```

To delete all proxy records by using the configuration utility

1. Navigate to Traffic Management > DNS > Records > Address Records.
2. In the details pane, click Flush Proxy Records.

Adding DNS Resource Records

Updated: 2013-08-26

You can add DNS records to a domain for which the Citrix® NetScaler® appliance is configured as a DNS proxy server. For information about adding DNS records, see [Configuring DNS Resource Records](#).

Removing a Load Balancing DNS Virtual Server

Updated: 2013-08-27

For information about removing a load balancing virtual server, see [Load Balancing](#).

Limiting the Number of Concurrent DNS Requests on a Client Connection

Updated: 2013-09-10

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 NetScaler 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.

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the command line interface

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

Example

```
> set dns parameter -maxPipeline 1000
Done
> show dns parameter
DNS parameters:
DNS retries: 5
.
.
.
```

Max DNS Pipeline Requests: 1000

Done

>

To specify the maximum number of concurrent DNS requests allowed on a single client connection by using the configuration utility

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.

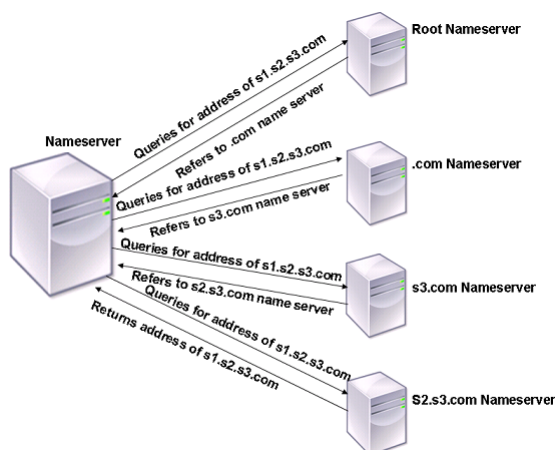
Configuring the NetScaler as an End Resolver

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



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.

This document includes the following information:

- [Enabling Recursive Resolution](#)
- [Setting the Number of Retries](#)

Enabling Recursive Resolution

Updated: 2013-08-27

To configure the NetScaler appliance to function as an end resolver, you must enable recursive resolution on the appliance.

To enable recursive resolution by using the command line interface

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

- `set dns parameter -recursion ENABLED`
- `show dns parameter`

Example

```
> set dns parameter -recursion ENABLED
Done
> show dns parameter
DNS parameters:
```

```

      .
      .
      .
Recursive Resolution : ENABLED
      .
      .
      .
Done
>

```

To enable recursive resolution by using the configuration utility

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.

Setting the Number of Retries

Updated: 2013-08-27

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.

To set the number of DNS retries by using the command line interface

At the command prompt, type the following commands to set the number of retries and verify the configuration:

- o set dns parameter -retries <positive_integer>
- o show dns parameter

Example

```

> set DNS parameter -retries 3
Done
> show dns parameter
    DNS parameters:
    DNS retries: 3
      .
      .
      .
Done
>

```

To set the number of retries by using the configuration utility

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.

Configuring the NetScaler as a Forwarder

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 as a forwarder, you must add an external name server (a name server other than the Citrix NetScaler appliance).

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 should forward name resolution queries. You can specify the lookup priority to specify the name service that the NetScaler appliance must use for name resolution.

Adding a Name Server

You can create a name server by specifying its IP address or by configuring an existing virtual server as the name server.

While adding name servers, you can provide an IP address or a virtual IP address (VIP). If you add an IP address, the NetScaler load balances requests to the configured name servers in round robin method. If you add a VIP, you can configure any load balancing method.

Example 1, which follows the command synopsis below, adds a local name server. Example 2 specifies the name of a load balancing virtual server of service type DNS.

Note: To verify the configuration, you can also use the `sh dns <recordtype> <domain>` command. If the queried records are not present in the cache, the resource records are fetched from the configured external name servers.

To add a name server by using the command line interface

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

- `add dns nameServer ((<IP> [-local]) | <dnsVserverName>)`
- `show dns nameServer [<IP> | <dnsVserverName>]`

Example 1

```
> add dns nameServer 10.102.9.20 -local
Done
> show dns nameServer 10.102.9.20
1)      10.102.9.20: LOCAL - State: UP
Done
>
```

Example 2

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

To remove a name server by using the NetScaler command line, at the NetScaler command prompt, type the `rm dns nameServer` command followed by the IP address of the name server.

To add a name server by using the configuration utility

Navigate to Traffic Management > DNS > Name Servers and create a name server.

Setting DNS Lookup Priority

You can set the lookup priority to either DNS or WINS. This option is used in the SSL VPN mode of operation.

To set the lookup priority to DNS by using the command line interface

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

Example

```
> set dns parameter -nameLookupPriority DNS
Done
> show dns parameter
.
.
.
Name lookup priority : DNS
.
.
.
Done
>
```

To set lookup priority to DNS by using the configuration utility

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.

Disabling and Enabling Name Servers

The following procedure describes the steps to enable or disable an existing name server.

To enable or disable a name server by using the command line interface

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>

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
>
```

To enable or disable a name server by using the configuration utility

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.

Configuring DNS Suffixes

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.

Creating 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.

To create DNS suffixes by using the command line interface

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>

Example

```
> add dns suffix example.com
Done
> show dns suffix example.com
1)      Suffix: example.com
Done
>
```

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.

To create DNS suffixes by using the configuration utility

Navigate to Traffic Management > DNS > DNS Suffix and create DNS suffixes.

DNS ANY Query

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

Updated: 2013-08-26

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](#).

Domain Name System Security Extensions

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 Citrix® NetScaler® ADC. You can generate and import keys for signing DNS zones. You can configure DNSSEC for zones for which the NetScaler ADC 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.

Configuring DNSSEC

Configuring DNSSEC involves enabling DNSSEC on the Citrix® NetScaler® 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 ADC does not act as a DNSSEC resolver. DNSSEC on the ADC is supported only in the following deployment scenarios:

1. ADNSâ€™NetScaler is the ADNS and generates the signatures itself.
2. Proxyâ€™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.

This document includes the following information:

- o [Enabling and Disabling DNSSEC](#)
- o [Creating DNS Keys for a Zone](#)
- o [Publishing a DNS Key in a Zone](#)
- o [Configuring a DNS Key](#)
- o [Signing and Unsigning a DNS Zone](#)
- o [Viewing the NSEC Records for a Given Record in a Zone](#)
- o [Removing a DNS Key](#)

Enabling and Disabling DNSSEC

Updated: 2014-08-27

You must enable DNSSEC on the NetScaler ADC 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 ADC to respond to clients with DNSSEC-specific information.

To enable or disable DNSSEC by using the command line interface

At the command prompt, type the following commands to enable or disable DNSSEC and verify the configuration:

- o `set dns parameter -dnssec (ENABLED | DISABLED)`
- o `show dns parameter`

Example

```
> set dns parameter -dnssec ENABLED
Done
> show dns parameter
DNS parameters:
DNS retries: 5
.
.
.
DNSSEC Extension: ENABLED
Max DNS Pipeline Requests: 255
Done
>
```

To enable or disable DNSSEC by using the configuration utility

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.

Creating DNS Keys for a Zone

Updated: 2014-10-29

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 ADC 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 [Publishing 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.

To create a DNS key by using the command line interface

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 -fileNa
File Name: /nsconfig/dns/example.com.zsk.rsasha1.1024.key (public); /nsconfig/dns/example.co
This operation may take some time, Please wait...
Done
> create dns key -zoneName example.com -keyType ksk -algorithm RSASHA1 -keySize 4096 -fileNa
File Name: /nsconfig/dns/example.com.ksk.rsasha1.4096.key (public); /nsconfig/dns/example.co
This operation may take some time, Please wait...
Done
>
```

To create a DNS key by using the configuration utility

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 compute 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.

Publishing a DNS Key in a Zone

Updated: 2014-10-29

A key (Zone Signing Key or Key Signing Key) is published in a zone by adding the key to the NetScaler ADC. 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 ADC (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.

To publish a key in a zone by using the command line interface

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>]

Example

```
> add dns key example.com.zsk example.com.zsk.rsasha1.1024.key example.com.zsk.rsasha1.1024.  
Done  
> add dns key example.com.ksk example.com.ksk.rsasha1.4096.key example.com.ksk.rsasha1.4096.  
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  
>
```

To publish a key in a DNS zone by using the NetScaler configuration utility

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.

Configuring a DNS Key

Updated: 2014-08-27

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 ADC automatically re-signs all the resource records in the zone with the key, provided that the zone is currently signed with the particular key.

To configure a key by using the command line interface

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>]

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
>
```

To configure a key by using the configuration utility

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:
 - Expiresâ€™expires
 - Notification Periodâ€™notificationPeriod
 - TTLâ€™TTL
4. Click OK.

Signing and Unsigning a DNS Zone

Updated: 2014-08-27

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 ADC 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.

To sign a zone by using the command line interface

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

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
>
save config
```

To unsign a zone by using the command line interface

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)]

Example


```

> unsigned 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
>

```

To sign or unsign a zone by using the configuration utility

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:
 - o 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.

- o 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.

Viewing the NSEC Records for a Given Record in a Zone

Updated: 2014-08-27

You can view the NSEC records that the NetScaler ADC automatically creates for each owner name in the zone.

To view the NSEC record for a given record in a zone by using the command line interface

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
>

```

To view the NSEC record for given record in a zone by using the configuration utility

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.

Removing a DNS Key

Updated: 2014-08-27

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.

To remove a key from the NetScaler ADC by using the command line interface

At the command prompt, type the following command to remove a key and verify the configuration:

- o `rm dns key <keyName>`
- o `show dns key <keyName>`

Example

```
> rm dns key example.com.zsk
Done
> show dns key example.com.zsk
ERROR: No such resource [keyName, example.com.zsk]
>
```

To remove a key from the NetScaler ADC by using the configuration utility

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.

Configuring DNSSEC When the NetScaler ADC is Authoritative for a Zone

When the Citrix® 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 [Creating DNS Keys for a Zone](#), [Publishing a DNS Key in a Zone](#), and [Signing and Unsigning 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 [Enabling and Disabling DNSSEC](#).

Finally, after you configure DNSSEC for the authoritative zone, you must save the NetScaler configuration.

Configuring DNSSEC for a Zone for Which the NetScaler ADC Is a DNS Proxy Server

The procedure for signing a zone for which the Citrix® 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 ADC 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.

This document includes the following information:

- [Configuring DNSSEC for a Zone-Less DNS Proxy Server Configuration](#)
- [Configuring DNSSEC for a Partial Zone Ownership Configuration](#)

Configuring DNSSEC for a Zone-Less DNS Proxy Server Configuration

Updated: 2014-08-27

For a zone-less DNS proxy server configuration, zone signing must be performed on the backend name servers. On the NetScaler ADC, 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 [Configuring 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.

Configuring DNSSEC for a Partial Zone Ownership Configuration

Updated: 2014-08-27

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 ADC. 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 ADC 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.

To 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.

To sign the partial zone on the NetScaler ADC

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 [Configuring 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 [Publishing 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 [Signing and Unsigning a DNS Zone](#).

Configuring DNSSEC for GSLB Domain Names

If global server load balancing (GSLB) is configured on the Citrix® 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 [Configuring 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 [Configuring DNSSEC for a Partial Zone Ownership Configuration](#).

Zone Maintenance

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 have to be performed manually. The process of re-signing a zone is performed automatically and does not require manual intervention.

This document includes the following information:

- [Re-Signing an Updated Zone](#)
- [Rolling Over DNSSEC Keys](#)

Re-Signing an Updated Zone

Updated: 2014-08-27

When a zone is updated, that is, when new records are added to the zone or existing records are changed, the process of re-signing the new (or modified) record is performed automatically by the Citrix® NetScaler® ADC. If a zone contains multiple Zone Signing Keys, the ADC re-signs the new (or modified) record with the key with which the zone is signed at the point in time when the re-signing is to be performed.

Rolling Over DNSSEC Keys

Updated: 2014-08-27

On the NetScaler ADC, 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 .

Pre-Publish Key Rollover

RFC 4641, “DNSSEC Operational Practices” defines four stages for the pre-publish 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 currently signed. The zone contains only those RRSIGs that were 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 (that corresponds to the new key) enough 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 new DNS key by using the `create dns key` command.

For more information about creating a DNS key, including an example, see [Creating 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 [Publishing a DNS Key in a Zone](#).

- o **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 that were 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 that were 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 unsigned a zone, including examples, see [Signing and Unsigning a DNS Zone](#).

- o **Stage 4: DNSKEY Removal.** When the RRSIGs that were 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 [Removing 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.

- o **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 currently signed. The zone contains only those RRSIGs that were generated by the `example.com.zsk1` key, which is due for expiry. The Key Signing Key is `example.com.ksk1`.

- o **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 new DNS key by using the `create dns key` command.

For more information about creating a DNS key, including an example, see [Creating 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 [Publishing 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 [Signing and Unsigning a DNS Zone](#).

- o **Stage 3: DNSKEY Removal.** When the RRSIGs that were 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 [Removing a DNS Key](#).

Offloading DNSSEC Operations to the NetScaler ADC

For DNS zones for which your DNS servers are authoritative, you can offload DNSSEC operations to the NetScaler ADC. 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 [Configuring 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 ADC. 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 [Enabling Caching of DNS Records](#).

Note: DNSSEC offloading is supported on all NetScaler MPX platforms, except the NetScaler MPX 9700/10500/12500/15500 FIPS platform. The feature is also supported on NetScaler virtual appliances hosted on NetScaler SDX platforms.

To enable DNSSEC offloading for a zone by using the command line interface

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

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
>
```

To enable DNSSEC offloading for a zone by using the configuration utility

1. Navigate to Traffic Management > DNS > Zones.
2. In the details pane, do one of the following:
 - To create a zone on the ADC, 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 ADC to generate NSEC records for the zone, select the NSEC check box.

DataStream

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 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.

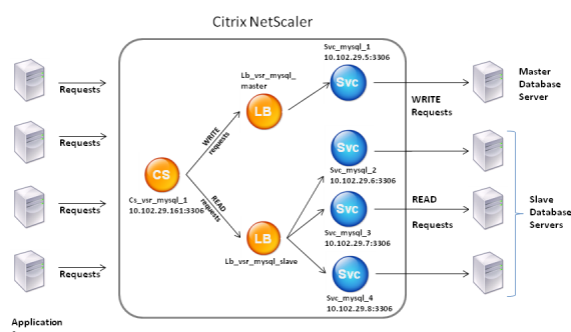
How NetScaler DataStream Works

Updated: 2013-09-18

In DataStream, the NetScaler is placed in-line between the application and/or Web servers and the database servers. On the NetScaler 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*



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 NetScaler, then, authenticates the clients using the database user credentials configured on the NetScaler 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 NetScaler) based on the load balancing algorithm. The NetScaler 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 NetScaler, the clients (application or Web servers) send their requests to the IP address of a load balancing virtual server configured on the NetScaler appliance. The NetScaler authenticates the client by using the database user credentials configured on the NetScaler 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

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 ADC. 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>
```

Example

```
> add db user nsdbuser -password dd260427edf
```

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.

To add a database user by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and enter new values for the password.

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.

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>
```

Example

```
> set db user nsdbuser -password dd260538abs
```

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. However, if the user continues to exist on the database server and you remove the user from the NetScaler, 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>
```

Example

```
> rm db user nsdbuser
```

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

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

Example

```
> add dbProfile myDBProfile -interpretQuery YES -stickiness YES -kcdAccount mykcdacctnt
Done
> show dbProfile myDBProfile
    Name: myDBProfile
    Interpret Query: YES
    Stickyness: YES
    KCD Account: mykcdacctnt
    Reference count: 0

Done
>
```

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>
```

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

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.

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](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

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.

Note: For information about default syntax expressions, see [Default Syntax Expressions: DataStream](#).

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](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

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. For information about configuring monitors, see [Monitors](#).

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.

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"
```

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"
```

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"
```

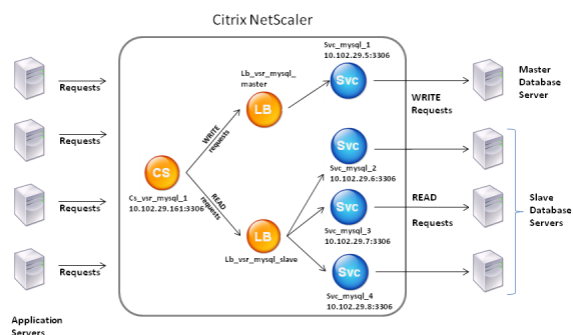
Use Case 1: Configuring DataStream for a Master/Slave Database Architecture

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*



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.

Table 1. *Entity and Policy Names and Values*

Entity Type	Name	IP Address	Protocol	Port	Expression
Services	Svc_mysql_1	10.102.29.5	MYSQL	3306	NA
^	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_master	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_1	10.102.29.161	MYSQL	3306	NA
Content switching policy	Cs_select	NA	NA	NA	"MYSQL.REQ.QUERY.COMMAND.contains(\"select\")"

To configure DataStream for a master/slave database setup by using the command line interface

At the command prompt, type

- o add service Svc_mysql_1 10.102.29.5 mysql 3306
- o add service Svc_mysql_2 10.102.29.6 mysql 3306
- o add service Svc_mysql_3 10.102.29.7 mysql 3306
- o add service Svc_mysql_4 10.102.29.8 mysql 3306
- o add lb vserver Lb_vsr_mysql_master mysql 10.102.29.201 3306
- o add lb vserver Lb_vsr_mysql_slave mysql 10.102.29.202 3306
- o bind lb vserver Lb_vsr_mysql_master svc_mysql_1
- o bind lb vserver Lb_vsr_mysql_slave svc_mysql_2
- o bind lb vserver Lb_vsr_mysql_slave svc_mysql_3
- o bind lb vserver Lb_vsr_mysql_slave svc_mysql_4
- o add cs vserver Cs_vsr_mysql_1 mysql 10.102.29.161 3306
- o add cs policy Cs_select "rule "MYSQL.REQ.QUERY.COMMAND.contains(\"select\")"
- o bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_master
- o bind cs vserver Cs_vsr_mysql_1 Lb_vsr_mysql_slave "policy Cs_select "priority 10

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

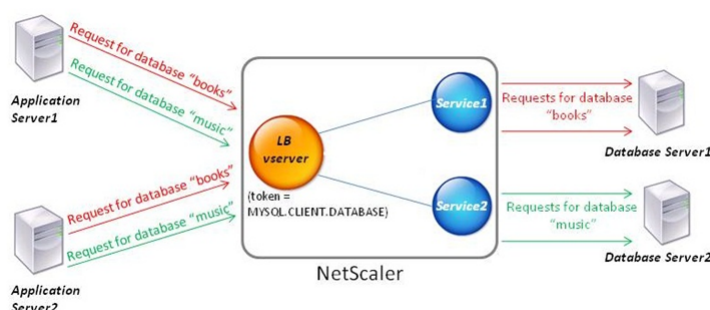
You can configure the token method of load balancing for DataStream to base the selection of database servers on the value of the token extracted from the client (application or web server) requests. These tokens are defined by using SQL expressions. For subsequent requests with the same token, the NetScaler 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	Â

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



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:

- o add service Service1 192.0.2.9 MYSQL 3306
- o add service Service2 192.0.2.11 MYSQL 3306
- o add lb vserver token_lb_vserver MYSQL 192.0.2.15 3306 -lbmethod token -rule MYSQL.CLIENT.DATABASE
- o bind lb vserver token_lb_vserver Service1
- o bind lb vserver token_lb_vserver Service2

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

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.

This section includes the following details:

- [Summary of Configuration Tasks](#)
- [Configuring Transparent Mode by Using a Wildcard Virtual Server](#)
- [Configuring Transparent Mode by Using MSSQL Services](#)

Summary of Configuration Tasks

Updated: 2015-05-25

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. For instructions, see [Configuring Transparent Mode by Using a Wildcard Virtual Server](#).
 - 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. For instructions, see [Configuring Transparent Mode by Using MSSQL Services](#).
- 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

Updated: 2013-11-04

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:

1. add lb vserver <name> <serviceType> <IPAddress> <port>
2. show lb vserver <name>

Example

```
> add lb vserver wildcardLbVs MSSQL * 1433
Done
> show lb vserver wildcardLbVs
wildcardLbVs (*:1433) - MSSQL          Type: ADDRESS
State: UP
. . .

Done
>
```

To create a wildcard virtual server by using the NetScaler configuration utility

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 command line

At the command prompt, type the following commands to enable USIP mode globally and verify the configuration:

- o enable ns mode USIP
- o show ns mode

Example

```
> enable ns mode USIP
Done
> show ns mode
```

	Mode	Acronym	Status
	-----	-----	-----
	. . .		
3)	Use Source IP	USIP	ON
	. . .		

```
Done
>
```

To enable USIP mode globally by using the NetScaler configuration utility

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

Updated: 2013-08-23

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:

- o add service <name> (<IP> | <serverName>) <serviceType> <port> -usip YES
- o show service <name>

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
>
```

To create an MSSQL service, with USIP enabled, by using the NetScaler configuration utility

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

A database server farm should be load balanced not only on the basis of the states of the servers, but also on the basis of 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. If a query is forwarded to a service on which the database is unavailable, the request is not served. 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, and 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, and 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 Citrix 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 currently 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 enable the load balancing feature, 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.

This section includes the following details:

- [How Database Specific Load Balancing Works](#)
- [Enabling Load Balancing](#)
- [Configuring a Load Balancing Virtual Server for Database Specific Load Balancing](#)
- [Configuring Services](#)
- [Configuring Database Users](#)
- [Configuring a Monitor to Retrieve the Names of Active Databases](#)
- [HA Group Deployment Support for MSSQL](#)

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 Citrix NetScaler appliance stores the results, and regularly updates the records on the basis of the information retrieved through monitoring. When a client queries a particular database, the appliance uses the configured load balancing method to select a service, and then checks its 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

Updated: 2013-08-08

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
```

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

Updated: 2014-05-30

To configure a virtual server to load balance databases on the basis of availability, you enable the database specific load balancing parameter on the virtual server. Enabling the parameter modifies the load balancing logic so that the NetScaler appliance refers the results of the monitoring probe sent to the selected service, before forwarding the query to that service.

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>

Example

In the following example, we create a Microsoft SQL virtual server.

```
> add lb vserver DBSpecificLB1 MSSQL 192.0.2.10 1433 -dbsLb ENABLED
Done
> show lb vserver DBSpecificLB1
      DBSpecificLB1 (192.0.2.10:1433) - MSSQL          Type: ADDRESS
      . . .
      DBS_LB: ENABLED
Done
>
```

Configuring Services

Updated: 2014-06-03

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.

Example

In the following example, we create a service of type MSSQL


```
add service msservice1 5.5.5.5 MSSQL 1433
```

Configuring Database Users

Updated: 2014-10-21

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 ADC. 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>
```

Example

```
> add db user nsdbuser -password dd260427edf
```

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.

To add a database user by using the configuration utility

Navigate to System > User Administration > Database Users, select a user, and enter new values for the password.

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.

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>
```

Example

```
> set db user nsdbuser -password dd260538abs
```

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. However, if the user continues to exist on the database server and you remove the user from the NetScaler, 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>
```

Example

```
> rm db user nsdbuser
```

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

Updated: 2014-05-30

To retrieve a list of all the active databases on a database instance, you create a monitor that 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 a 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
```

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, and 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>

Example

In the following example, we create an MSSQL-ECV type monitor.

```
> 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 ENABLED
Done
> 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
>
```

To configure a monitor to retrieve the names of all the active databases hosted on a service by using the configuration utility

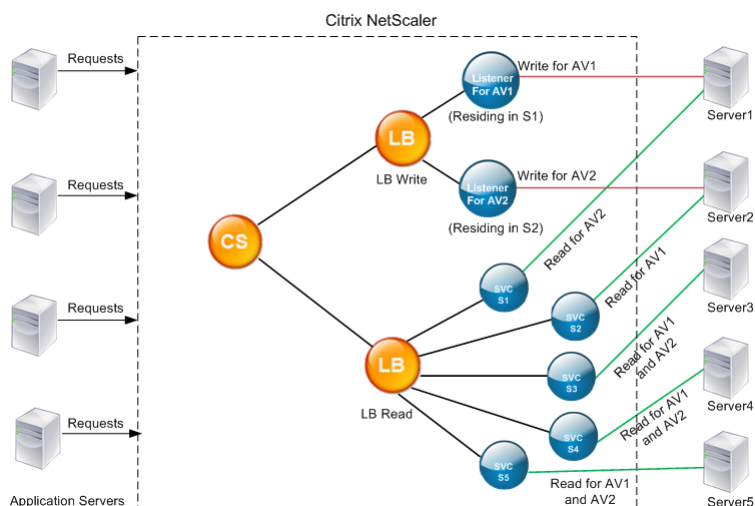
1. Navigate to Traffic Management > Load Balancing > Monitors and configure a monitor of type MSSQL-ECV or MYSQL-ECV.
2. In Special Parameters, specify a user name, query (for example "select name from sys.databases where state=0"), and a rule (for example MSSQL.RES.TYPE.NE(ERROR)).

HA Group Deployment Support for MSSQL

Updated: 2014-06-17

Consider the following scenario in which database specific load balancing is configured in a high availability group deployment. S1 through S5 are the services on the NetScaler. DB1 through DB4 are 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. AV1 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.



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.
 - o add lb vserver lbwrite -dbslb enabled
 - o add lbvserver lbread MSSQL -dbslb enabled
 - o 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.
 - o add service L1 1.1.1.11 MSSQL 1433
 - o add service L2 1.1.1.12 MSSQL 1433
 - o add service s1 1.1.1.13 MSSQL 1433
 - o add service s2 1.1.1.14 MSSQL 1433
 - o add service s3 1.1.1.15 MSSQL 1433
 - o add service s4 1.1.1.16 MSSQL 1433
 - o add service s5 1.1.1.17 MSSQL 1433
3. Bind the services to the load balancing virtual servers.
 - o bind lbvserver lbwrite L1
 - o bind lbvserver lbwrite L2
 - o bind lbvserver lbread s1
 - o bind lbvserver lbread s2
 - o bind lbvserver lbread s3
 - o bind lbvserver lbread s4
 - o bind lbvserver lbread s5
4. Configure database users.
 - o add db user nsdbuser1 -password dd260427edf

- o 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.
 - o add lb monitor monitor_L1 MSSQL-ECV -userName user1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states 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_addresses 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
 - o add lb monitor monitor_L2 MSSQL-ECV -userName user1 -sqlQuery "SELECT name FROM sys.databases a INNER JOIN sys.dm_hadr_availability_replica_states 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_addresses 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
 - o add lb monitor monitor1 MSSQL-ECV -userName user1 -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.
 - o add cs policy pol_write -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(\"insert\")"
 - o add cs policy pol_read -rule "MSSQL.REQ.QUERY.TEXT.CONTAINS(\"select\")"
- 7. Bind the policies to the content switching virtual server.
 - o bind csvserver csv -targetLBVserver lbwrite -policyName pol_write -priority 11
 - o 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.
 - o bind service L1 -monitorName monitor_L1
 - o bind service L2 -monitorName monitor_L2
 - o bind service s1 -monitorName monitor1
 - o bind service s2 -monitorName monitor1
 - o bind service s3 -monitorName monitor1
 - o bind service s4 -monitorName monitor1
 - o bind service s5 -monitorName monitor1

DataStream Reference

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 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

Updated: 2014-08-28

Â	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, 2008R2, and 2012.
Protocols	MySQL protocol version 10. For information about the MySQL protocol, see http://dev.mysql.com/doc/internals/en/client-server-protocol.html	TDS protocol version 7.1 and higher For information about the TDS protocol, see http://msdn.microsoft.com/en-us/library/dd30452
Authentication Methods	MySQL native authentication is supported.	SQL server authentication and Windows Authentication (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.

NetScaler 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>

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.

NetScaler 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. NetScaler 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 NetScaler begins connection multiplexing.

Special Queries

Updated: 2014-05-21

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, NetScaler 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 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, NetScaler 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 NetScaler 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, NetScaler is unable to differentiate between local and global variables. For this query, the NetScaler uses the 'store and forward' mechanism described earlier .

USE <db> query

Using this query, the user can change the database associated with a connection. In this case, NetScaler 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, NetScaler parses the <init_db> value sent and modifies a field in the server side connection to reflect the new database to be used.

COM_PREPARE

NetScaler 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 NetScaler uses the 'store and forward' mechanism described earlier in this section.

Audit Log Message Support

Updated: 2013-09-30

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.

Firewall Load Balancing

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 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 for the destination IP address and enable L3 mode, the NetScaler 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 with IP address and port as * and 21 respectively, and the service type specified as FTP. In this case, the NetScaler 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. 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

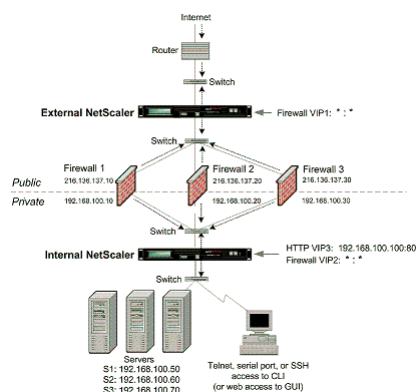
A NetScaler deployment in a sandwich mode is capable of load balancing network traffic through firewalls in both directions: ingress (traffic entering the network from the outside, such as the internet) and egress (traffic leaving the network to the internet).

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)



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

Updated: 2015-05-22

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

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			

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 *
```

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

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>

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
```

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 * *
```

Example

```
add lb vserver Vserver-LB-1 ANY * *
```

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>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

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>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

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:

- o save ns config
- o show vserver

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

```

Configuring the Internal NetScaler ADC in a Sandwich Environment

Updated: 2015-06-04

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:

- o enable ns feature LB
- o 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			

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 *
```

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

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:

- o add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- o bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

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 * *
```

Example

```
add lb vserver Vserver-LB-1 ANY * *
```

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:
 - o 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>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

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>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

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>
```

Example

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
```

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**.

- 1.

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>
```

Example

```
bind mon monitor-HTTP-1 Service-HTTP-1
```

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.

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>
```

Example

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```

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.

1.

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 → IP Address
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

```

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

Updated: 2013-09-05

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:

- o Name
- o IP address
- o Port
- o Protocol
- o State of the virtual server
- o Rate of requests received
- o 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>]
```

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
BRVSERVER	10.10.1.1	80	HTTP	DOWN	0/s	0/s
LBVIP	10.102.29.66	80	HTTP	UP	0/s	0/s

```

Done

```

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>

Example

```
stat service Service-HTTP-1
```

To view the statistics of a service by using the configuration utility

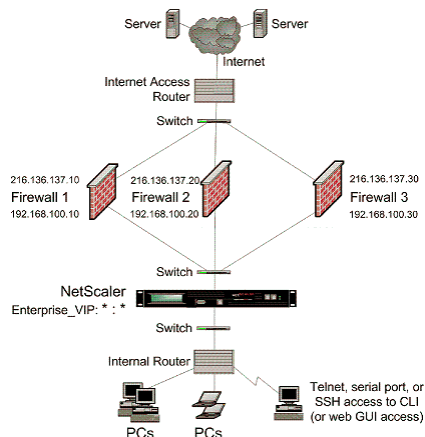
1. Navigate to Traffic Management > Load Balancing > Services > Statistics.
2. If you want to display the statistics for only one service, select the service, and click Statistics.

Enterprise Environment

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)



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

Updated: 2013-11-08

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			

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 *
```

Example

```
add service Service-HTTP-1 10.102.29.5 ANY *
```

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:
 - o Service Nameâ€”name
 - o Serverâ€”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:

- o add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- o bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

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:

- o Name*
- o Type*â€”type
- o Destination IP
- o 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 * *
```

Example

```
add lb vserver Vserver-LB-1 ANY * *
```

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:
 - o Nameâ€”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>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

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>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

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>
```

Example

```
add service Service-HTTP-1 10.102.29.5 HTTP 80
```

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 Nameâ€”name
 - Serverâ€”serverName
 - Portâ€”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>
```

Example

```
bind mon monitor-HTTP-1 Service-HTTP-1
```

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>
```

Example

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```

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:
 - Nameâ€”name
 - IP Addressâ€”IP Address
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**).
 - Portâ€”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>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

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:

- o save ns config
- o 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 millisc
```

Done

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

Updated: 2013-09-05

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:

- o Name
- o IP address
- o Port
- o Protocol
- o State of the virtual server
- o Rate of requests received
- o 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>]
```

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
BRV SERV	10.10.1.1	80	HTTP	DOWN	0/s	0/s
LBVIP	10.102.29.66	80	HTTP	UP	0/s	0/s

Done

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>
```

Example

```
stat service Service-HTTP-1
```

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.

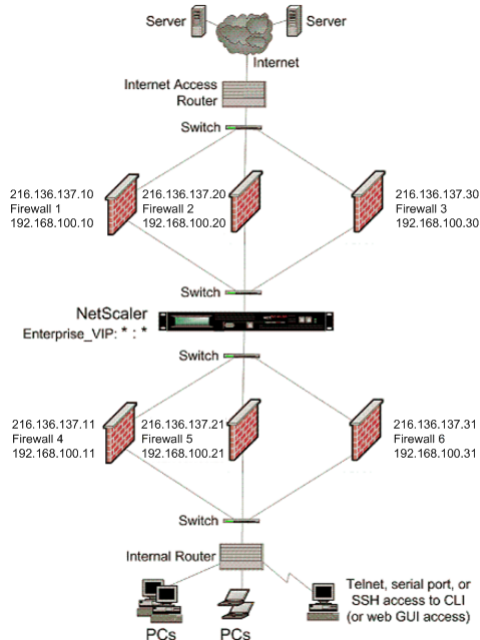
Multiple-Firewall Environment

Note: This feature is available in NetScaler release 9.3.e and 10.

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. 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.

Figure 1 shows a multiple-firewall load balancing environment

Figure 1. Firewall Load Balancing (multiple-firewall)



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

Updated: 2015-05-18

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. 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 command line interface

At the command prompt, type the following command to enable load balancing and verify the configuration:

- o enable ns feature <featureName>
- o 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
```

To enable load balancing by using the configuration utility

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 command line interface

At the command prompt, type the following command to configure support for all the protocols and ports:

```
add service <name>@ <serverName> <serviceType> <port_number>
```

Example

```
add service fw-svc1 10.102.29.5 ANY *
```

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 Services dialog box, specify values for the following parameters as shown:

- o Service Name "name"
- o 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 command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- o add lb monitor <monitorName> <type> [-destIP <ip_addr|ipv6_addr|*>] [-transparent (YES | NO)]
- o bind lb monitor <monitorName> <serviceName>

Example

```
add monitor monitor-HTTP-1 HTTP -destip 10.10.10.11 -transparent YES
bind monitor monitor-HTTP-1 fw-svc1
```

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 for the following parameters as shown:

- o Name*
- o Type*â€™type
- o Destination IP
- o 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 command line interface

At the command prompt, type the following command:

```
add lb vserver <name>@ <serviceType> <IPAddress> <port_number>
```

Example

```
add lb vserver Vserver-LB-1 ANY * *
```

To configure a 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 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 command line interface

At the command prompt, type the following command:

```
set lb vserver <name>@ -m <RedirectionMode>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

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-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, you need to bind it to a wildcard virtual server.

To bind firewall services to the virtual server by using the command line interface

At the command prompt, type the following command:

```
bind lb vserver <name>@ <serviceName>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind firewall services to the virtual server 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-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 multipl-firewall load balancing by using the vServerSpecificMac parameter.

To configure multiple-firewall load balancing by using the command line interface

At the command prompt, type the following command:

```
set lb parameter -vServerSpecificMac <status>
```

Example

```
set lb parameter -vServerSpecificMac ENABLED
```

To configure multiple-firewall load balancing 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, 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 command line interface

At the command prompt, type the following commands to configure a transparent monitor and verify the configuration:

- o save ns config
- o 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

```

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 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

Updated: 2013-09-05

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:

- o Name
- o IP address
- o Port
- o Protocol
- o State of the virtual server
- o Rate of requests received
- o 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>]
```

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
BRV SERV	10.10.1.1	80	HTTP	DOWN	0/s	0/s
LBVIP	10.102.29.66	80	HTTP	UP	0/s	0/s

```
Done
```

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>
```

Example

```
stat service Service-HTTP-1
```

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.

Global Server Load Balancing

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.

Following are some typical GSLB configurations:

- **Active-active data center setup.** Consists of multiple active data centers. Client requests are load balanced across active data centers.
- **Active-standby data center setup.** 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.
- **Proximity setup.** Directs client requests to the data center that is closest in geographical distance or network distance.

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

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.

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.

This document includes the following information:

- **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.

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.

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

Updated: 2013-09-13

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.

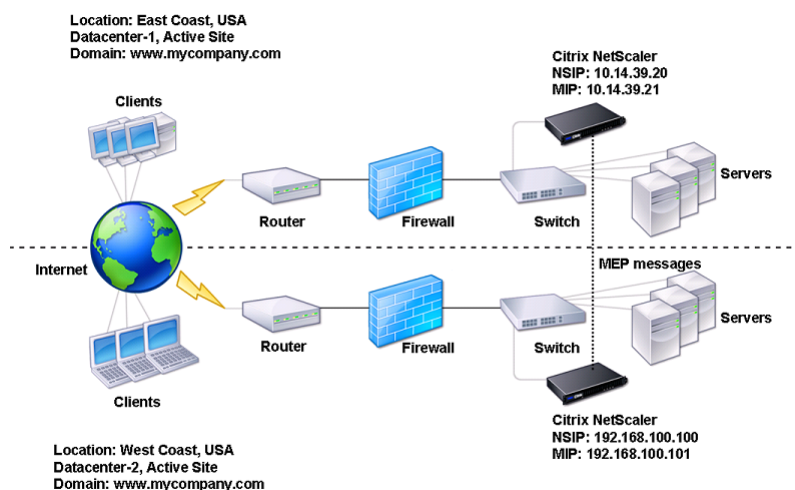
DNS VIPs

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)

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



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 GSLB 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

Updated: 2013-08-30

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

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 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:

- o add service <name> <IP>@ ADNS <port>
- o show service <name>

Example

```
add service Service-ADNS-1 10.14.39.21 ADNS 53
show service Service-ADNS-1
```

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>
```

Example

```
set service Service-ADNS-1 10.14.39.21 ADNS 53
```

To remove an ADNS service by using the command line interface

At the command prompt, type the following command:

```
rm service <name>
```

Example

```
rm service Service-ADNS-1
```

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

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:

- o add gslb site <siteName> <siteIPAddress>
- o show gslb site <siteName>

Example

```
add gslb site Site-GSLB-East-Coast 10.14.39.21
show gslb site Site-GSLB-East-Coast
```

To modify or remove a GSLB Site by using the command line interface

- o 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.
- o 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.
- o 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>
```

Example

```
stat gslb site Site-GSLB-East-Coast
```

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

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.

Creating GSLB Services

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:

- o add gslb service <serviceName> <serverName | IP> <serviceType> <port>-siteName <string>
- o show gslb service <serviceName>

Example

```
add gslb service Service-GSLB-1 10.14.39.14 HTTP 80 "siteName Site-GSLB-East-Coast"
show gslb service Service-GSLB-1
```

To modify or remove a GSLB service by using the command line interface

- o 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
```

- o 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
```

- o 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>
```

Example

```
stat gslb service Service-GSLB-1
```

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

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.

Creating GSLB Virtual Servers

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>

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
```

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, which is just like using the add gslb vserver command, except that you enter the name of an existing GSLB virtual server.
- To reset a parameter to its 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>
```

Example

```
stat gslb vserver Vserver-GSLB-1
```

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 on the Statistics link from the configuration utility, the following details of the service will be 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

Note: This feature is available in NetScaler release 10.5.e.

You can now clear the statistics of a GSLB virtual server and service. NetScaler ADC provides the following two options to clear the statistics:

- o **Basic:** Clears the statistics that are specific to the virtual server but retains the statistics that are contributed by the bound GLSB service.
- o **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>
```

Example

```
stat gslb vserver Vserver-GSLB-1 â€"clearstats basic
```

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>
```

Example

```
stat gslb service service-GSLB-1 â€"clearstats basic
```

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

Updated: 2014-11-21

When you create a GSLB virtual server, it is enabled by default. If you disable it, it cannot process traffic. A disabled GSLB virtual server is not included in GSLB configuration but is not removed from the NetScaler appliance.

To enable or disable a GSLB virtual server by using the command line interface

At the command prompt, type one of the following commands:

- o enable gslb vserver <name>@
- o disable gslb vserver <name>@

Example

```
enable gslb vserver Vserver-GSLB-1  
disable gslb vserver Vserver-GSLB-1
```

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

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>`

Example

```
bind gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1
show gslb vserver Vserver-GSLB-1
```

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>
```

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

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 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:

- o `bind gslb vserver <name> -domainName <string>`
- o `show gslb vserver <name>`

Example

```
bind gslb vserver Vserver-GSLB-1 -domainName www.mycompany.com
show gslb vserver Vserver-GSLB-1
```

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>`

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:
 - o To create a new Domain, click Add.
 - o 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:
 - o Domain Name*â€”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>`

Example

```
stat gslb domain www.mycompany.com
```

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 Statistics.

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>

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

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

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.
- Synchronization is done only on the parent sites. If a GSLB site is configured as a child site, its configuration is not affected by synchronization.

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). 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 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.

Before you start the synchronization of a GSLB setup, make sure that:

- On all the GSLB sites including the master node, management access 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"](#) and ["Configuring NetScaler-Owned IP Addresses"](#).

- 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.

- o 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:

- o `sync gslb config [-preview | -forceSync <string> | -nowarn | -saveconfig] [-debug]`
- o `show gslb syncStatus`

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
```

To synchronize a GSLB configuration by using the configuration utility

1. Navigate to Traffic Management > GSLB and, under GSLB Configuration, click **Synchronize configuration on remote sites** and synchronize the GSLB configuration.

Configuring the Metrics Exchange Protocol (MEP)

The data centers in a GSLB setup exchange metrics with each other through the metrics exchange protocol (MEP), which is a proprietary protocol for the Citrix NetScaler. 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 metrics 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) or a mapped IP address (MIP) to establish a connection to the IP address of a different data center. However, you can configure a specific SNIP, MIP, the NetScaler IP address (NSIP), or a virtual IP address (VIP) 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 cannot configure a GSLB site IP address as the source IP address for site metrics exchange.

If the source and target sites for a MEP connection (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. 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 metrics exchange is enabled, the monitor controls the health status. Binding the monitors to the remote service allows the NetScaler to interact with a non-NetScaler load balancing device. The NetScaler can monitor non-NetScaler devices but cannot perform load balancing on them. The NetScaler can monitor non-NetScaler devices, and can perform load balancing on them if 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.

This document includes the following information:

- [Configuring Site Metric Exchange](#)
- [Configuring Network Metric Information Exchange](#)
- [Configuring Persistence Information Exchange](#)

Configuring Site Metric Exchange

Updated: 2014-11-24

Site metrics exchanged between the GSLB sites include the status of each load balancing and 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 metric 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>`

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
```

To enable or disable site metric exchange by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites, select the site.
2. In the **Configure GSLB Site** dialog box, select the **Metric Exchange** option.

Configuring Network Metric Information Exchange

Updated: 2014-11-24

You can enable or disable the exchange of round trip time (RTT) information about the client's local DNS when the GSLB dynamic method (RTT) is enabled. 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 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:

- o `set gslb site <siteName> -nwmetricExchange (ENABLED|DISABLED)`
- o `show gslb site <<siteName>`

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
```

To enable or disable network metric information exchange by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites.
2. In the **Configure GSLB Site** dialog box, select the **Network Metric Exchange** option.

Configuring Persistence Information Exchange

Updated: 2014-11-24

You can enable or disable the exchange of persistence information at each site. This information is exchanged every 5 seconds between NetScaler appliances participating in GSLB.

For details about configuring persistence, see "[Configuring Persistent Connections](#)".

To enable/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:

- o `set gslb site <siteName> -sessionExchange (ENABLED|DISABLED)`
- o `show gslb site <siteName>`

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
```

To enable/disable persistence information exchange by using the configuration utility

1. Navigate to Traffic Management > GSLB > Sites, double-click the site.
2. In the **Configure GSLB Site** dialog box, select the **Persistence Session Entry Exchange** option.

Configuring Site-to-Site Communication

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 either a subnet IP (SNIP) address or a mapped IP (MIP) 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](#)
- [Encrypting the Exchange of Site Metrics](#)
- [Configuring the Source IP Address for an RPC Node](#)

Changing the Password of an RPC Node

Updated: 2014-11-21

You can 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`

Example

```
> set ns rpcNode 192.0.2.4 -password mypassword
Done
> show ns rpcNode
.
.
.
2)      IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e
      SrcIP:      *      Secure:  OFF
Done
>
```

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 NetScaler command line, 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.

Encrypting the Exchange of Site Metrics

Updated: 2014-11-24

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:

- o set ns rpcNode <IPAddress> [-secure (YES | NO)]
- o show rpcNode

Example

```
> set ns rpcNode 192.0.2.4 -secure YES
Done
>
> show rpcNode
.
.
.
3)      IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:  192.0.2.3      Secure
Done
>
```

To unset the secure parameter by using the command line interface

To unset the secure parameter by using the NetScaler command line, 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**.

Configuring the Source IP Address for an RPC Node

Updated: 2014-11-24

By default, the NetScaler appliance uses a NetScaler-owned subnet IP (SNIP) address or mapped IP (MIP) address as the source IP address for an RPC node, but you can configure the appliance to use a specific SNIP address or MIP address. If neither a SNIP address nor a MIP address is available, the GSLB site cannot communicate with other sites. In such a scenario, you must configure either the NetScaler IP (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 or a MIP address.

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:

- o set ns rpcNode <IPAddress> [-srcIP <ip_addr|ipv6_addr|*>]
- o show ns rpcNode

Example

```
> set ns rpcNode 192.0.2.4 -srcIP 192.0.2.3
Done
> show ns rpcNode
.
.
.
2)      IPAddress:  192.0.2.4 Password:  d336004164d4352ce39e SrcIP:  192.0.2.3      Sec
Done
>
```

To unset the source IP address parameter by using the command line interface

To unset the source IP address parameter by using the NetScaler command line, type the unset rpcNode command, 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**.

Customizing Your GSLB Configuration

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.

This document includes the following information:

- [Modifying Maximum Connections or Maximum Bandwidth for a GSLB Service](#)
- [Creating CNAME-Based GSLB Services](#)
- [Configuring Transition Out-Of-Service State \(TROFS\) in GSLB](#)
- [Configuring Dynamic Weights for Services](#)

Modifying Maximum Connections or Maximum Bandwidth for a GSLB Service

Updated: 2014-11-26

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>`

Example

```
set gslb service Service-GSLB-1 â€"maxBandwidth 100 â€"maxClients 100
show gslb service Service-GSLB-1
```

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 Clientsâ€"maxClients
 - Max Bandwidthâ€"maxBandwidth

Creating CNAME-Based GSLB Services

Updated: 2014-11-24

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 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>
```

Example

```
add gslb service Service-GSLB-1 -cnameEntry transport.mycompany.com -siteName Site-GSLB-East
add gslb service Service-GSLB-2 -cnameEntry finance.mycompany.com -siteName Site-GSLB-West-C
```

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**.

Configuring 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.

Configuring Dynamic Weights for Services

Updated: 2015-06-02

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:

- o **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.
- o **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.
- o **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
```

Example

```
set gslb vserver vserver-GSLB-1 -dynamicWeight SERVICECOUNT
```

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 Method

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 NetScaler 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)

Updated: 2013-11-11

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 command line interface

At the command prompt, type:

```
set gslb vserver <name> -lbMethod GSLBMethod
```

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod ROUNDROBIN
```

To change the GSLB method by using the configuration utility

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

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](#)

Adding a Location File to Create a Static Proximity Database

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).

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: 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.

The following abbreviations are used in this section:

- **CSHN**. Short name of a country based on the country code standard of ISO-3166.
- **LCN**. Long name of the country.
- **RC**. Region code based on ISO-3166-2 (for US and Canada). The region code "FIPS-10-4" is used for the other regions.

Note: 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 command line interface

At the command prompt, type:

- `add locationFile <locationFile> [-format <format>]`
- `show locationFile`

Example

```
> add locationFile /var/nsmapi/locdb/nsgeo1.0 -format netscaler
Done
> show locationFile
Location File: /var/nsmapi/locdb/nsgeo1.0
Format: netscaler
Done
>
```

To add a static location file by using the configuration utility

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 NetScaler command line equivalent.

To view a static location file by using the configuration utility

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 nsmmap 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 NetScaler command prompt, type the following command:

```
nsmmap -f <inputFileFormat> -o <outputFileName> <inputFileName>
```

Example

```
nsmmap -f ip-country-region-city -o nsfile.ns ip-country-region-city.csv
```


Adding Custom Entries to a Static Proximity Database

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:

- o add location < IPfrom> < IPto> <preferredLocation> [-longitude <integer>][-latitude <integer>]]
- o show location

Example

```
>add location 192.168.100.1 192.168.100.100 *.us.ca.mycity
>show location
```

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.

Setting the Location Qualifiers

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. The 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>]
```

Example

```
set locationparameter -context custom -q1label asia
```

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.

Specifying the Proximity Method

When you have configured the static proximity database, you are ready to specify static proximity as the GSLB 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>`

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod STATICPROXIMITY
show gslb vserver
```

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**.

Synchronizing GSLB Static Proximity Database

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
```

Configuring the 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 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 Netscaler 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.

This document includes the following information:

- [Configuring a GSLB Virtual Server for Dynamic RTT](#)
- [Setting the Probing Interval of Local DNS Servers](#)

Configuring a GSLB Virtual Server for Dynamic RTT

Updated: 2014-11-24

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>
```

Example

```
set gslb vserver Vserver-GSLB-1 -lbMethod RTT -tolerance 10
```

To configure a GSLB virtual server for dynamic RTT by using the configuration utility

1. Navigate to Traffic Management > GSLB > Virtual Servers and double-click the virtual server.

Setting the Probing Interval of Local DNS Servers

Updated: 2014-11-24

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>
```

Example

```
set lb monitor monitor-HTTP-1 HTTP -interval 10 sec -resptimeout 5 sec
```

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).

Configuring Persistent Connections

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. Persistence is useful for deployments that deal with e-commerce, such as shopping card usage, where the server needs to maintain the state of the connection to track the transaction. To maintain the state of connection, you must configure persistence on a virtual server. With persistence configured, NetScaler selects a data center to process a client request and forwards the IP address of the selected data center for all subsequent DNS requests. If the configured persistence applies to a site that is down, the NetScaler appliance uses a GSLB method to select a new site, and the new site becomes persistent for subsequent requests from the client.

The GSLB virtual server is responsible for DNS-based site persistence, and it controls the site persistence for a remote GSLB service. The NetScaler appliance supports persistence based on the source IP address or on HTTP cookies.

When you bring a physical service DOWN with a delay time, the physical service goes into the transition out of service (TROFS) state. Site persistence is supported as long as the service is in the 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.

Note: If connection proxy is specified as the site persistence method and if you also want to configure persistence of the physical servers, do not configure SOURCEIP persistence. When the connection is proxied, an IP address owned by the NetScaler is used, and not the actual IP address of the client. Configure methods such as cookie persistence or rule-based persistence on the load balancing virtual server.

This document includes the following information:

- [Configuring Persistence Based on Source IP Address](#)
- [Configuring Persistence Based on HTTP Cookies](#)

Configuring Persistence Based on Source IP Address

Updated: 2014-11-24

With source-IP persistence, when a DNS request is received at a data center, the NetScaler appliance first looks for an entry in the persistence table and, if an entry for the local DNS server exists and the server mentioned in the entry is configured, the IP address of that server is sent as the DNS response.

For the first request from a particular client, the NetScaler appliance selects the best GSLB site for the request and sends its IP address to the client. Since persistence is configured for the source IP address of the client, all subsequent requests by that client or another local DNS server in the same IP subnet are sent the IP address of the GSLB site that was selected for the first request.

For source-IP address based persistence, the same set of persistence identifiers must be configured on the GSLB virtual servers in all data centers. A persistence identifier is a number used by the data centers to identify a particular GSLB virtual server. A cookie transmits the persistence identifier, enabling the NetScaler appliance to identify the domain so that it can forward all appropriate requests to the same domain. When persistence is enabled, the persistence information is also exchanged as part of metrics exchange.

For the NetScaler appliance to support persistence across sites, persistence must be enabled on the GSLB virtual servers of all participating sites. When you use source IP address persistence on the network identifier, you must configure a subnet mask. For any domain, persistence takes precedence over any other configured GSLB method.

To configure persistence based on source IP address by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -persistenceType (SOURCEIP|NONE) -persistenceId <positive_integer> [-persistMask <netmask>]
&#34;[timeout <mins>]
```

Example

```
set gslb vserver vserver-GSLB-1 -persistenceType SOURCEIP -persistenceId 23 -persistMask 255
```

To configure persistence based on source IP address by using the configuration utility

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:
 - o Persistence Idâ€œpersistenceID
 - o Time-outâ€œtimeout
 - o IPv4 Netmask or IPv6 Mask lengthâ€œpersistMask

Configuring Persistence Based on HTTP Cookies

Updated: 2014-11-26

The NetScaler appliance provides persistence at the HTTP-request level by using connection proxy and HTTP redirect. With these persistence methods, the appliance uses an HTTP cookie (known as a â€œsite cookieâ€œ) to reconnect the client to the same server. The NetScaler inserts the site cookie in the first HTTP response.

The site cookie contains information about the selected GSLB service on which the client has a persistent connection. 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.

When the NetScaler 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. The physical server in that GSLB site adds a site cookie to the HTTP header, and connection persistence is in effect.

If the DNS entry in the client cache expires, and then the client sends another DNS query and is directed to a different GSLB site, the new GSLB site uses the site cookie present in the client request header to implement persistence. If the GSLB configuration at the new site uses connection-proxy persistence, the new site creates a connection to the GSLB site that inserted the site cookie, proxies the client request to the original site, receives a response from the original GSLB site, relays that response back to the client, and closes the connection. If the GSLB configuration uses HTTP redirect persistence, the new site redirects the request to the site that originally inserted the cookie.

Note: Connection proxy persistence can be configured only for local services. However, connection proxy persistence must be enabled on both local and remote GSLB services that are configured for the GSLB virtual server. Connection proxy occurs when the following conditions are satisfied:

- o Requests are sent from a domain participating in GSLB. The domain is obtained from the URL/Host header.
- o Requests are sent from a local GSLB service whose public IP address matches the public IP address of an active service bound to the GSLB virtual server.
- o The local GSLB service has connection proxy enabled.
- o The request includes a valid cookie that contains the IP address of an active remote GSLB service.

If one of the conditions is not met, connection proxy does not occur, but a site cookie is added if the local GSLB service has connection proxy enabled AND:

- o No site cookie is supplied; OR,
- o The site cookie refers to an IP address that is not an active GSLB remote service; OR,
- o The cookie refers to the IP address of the virtual server on which the request is received.

The following are the limitations of using connection proxy site cookies:

- o Site cookies do not work for non-HTTP(S) protocols.
- o If an HTTP request is sent to a back-up virtual server, the virtual server does not add a cookie.
- o Site cookies do not work if SSL client authentication is required.
- o At the local site, the statistics for a GSLB service on a remote site are not the same as the statistics recorded for that service at the remote site. At the local site, the statistics for a remote GSLB service are slightly higher than the statistics that the remote site records for that same service.

Redirect persistence can be used only:

- o For HTTP or HTTPS protocols.
- o If the domain name is present in the request (either in the URL or in the HOST header), and the domain is a GSLB domain.
- o When the request is received on a backup VIP or a GSLB local service that is in the down state.

To set persistence based on HTTP cookies by using the command line interface

At the command prompt, type:

```
set gslb service <serviceName> -sitePersistence (ConnectionProxy [-sitePrefix <prefix>] | HTTPRedirect -sitePrefix <prefix>)
```


Example

```
set gslb service service-GSLB-1 -sitePersistence ConnectionProxy
set gslb service service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
```

To set persistence based on cookies by using the configuration utility

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.

Overriding Static Proximity Behavior by Configuring Preferred Locations

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 location 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"  
Done
```

```
> add dns policy dnsPolPrefLoc "CLIENT.IP.SRC.MATCHES_LOCATION(\"*.ZMB.*.*\")" prefLoc11
Done
```

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

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 NetScaler 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.

This document includes the following information:

- [Adding or Removing Monitors](#)
- [Binding Monitors to a GSLB Service](#)

Adding or Removing Monitors

Updated: 2014-11-24

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>`

Example

```
add lb monitor monitor-HTTP-1 -type HTTP -destPort 80
show lb monitor monitor-HTTP-1
```

To remove a monitor by using the command line interface

At the command prompt, type:

```
rm lb monitor <monitorName>
```

To add a monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and add or delete a monitor.

Binding Monitors to a GSLB Service

Updated: 2014-11-24

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>
```

Example

```
bind monitor monitor-HTTP-1 service-GSLB-1 -state enabled -weight 2
```

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>
```

Example

```
set gslb service service-GSLB-1 -monThreshold 9
```

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

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> â€"triggerMonitor (ALWAYS | MEPDOWN | MEPDOWN_SVCDOWN)
```

Example

```
> set gslb site Site-GSLB-North-America â€"triggerMonitor Always
Done
```

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

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.

This document includes the following information:

- [Configuring a Backup GSLB Virtual Server](#)
- [Configuring a GSLB Setup to Respond with Multiple IP Addresses](#)
- [Configuring a GSLB Virtual Server to Respond with an Empty Address Record When DOWN](#)
- [Configuring a Backup IP Address for a GSLB Domain](#)
- [Diverting Excess Traffic to a Backup Virtual Server](#)

Configuring a Backup GSLB Virtual Server

Updated: 2015-05-04

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), or you can configure a timeout period after which the primary takes over.

If you configure both the timeout and the `disablePrimaryOnDown` option for the backup entity, the backup session timeout takes precedence over the `disablePrimaryOnDown` setting.

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> [-backupSessionTimeout <timeoutValue>] [-disablePrimaryOnDown (ENABLED | DISABLED)]`
- `show gslb vserver <name>`

Example

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2 -backupSessionTimeout 3 -disab
show gslb vserver vserver-GSLB-1
```

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

Updated: 2014-11-24

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)`

- o `show gslb vserver <name>`

Example

```
set gslb vserver vserver-GSLB-1 -MIR ENABLED
show gslb vserver <vserverName>
```

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

Updated: 2014-11-24

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)
```

Example

```
> set gslb vserver vserver-GSLB-1 -EDR ENABLED
Done
```

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

Updated: 2014-11-24

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:

- o `set gslb vserver <name> -domainName <string> -backupIP <IPAddress>`
- o `show gslb vserver <name>`

Example

```
set gslb vserver vserver-GSLB-1 -domainName www.abc.com -backupIP 10.102.29.66
show gslb vserver vserver-GSLB-1
```

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

Updated: 2014-11-24

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 either by the number of connections or by the bandwidth.

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>`

- o show gslb vserver <name>

Example

```
set gslb vserver Vserver-GSLB-1 -soMethod CONNECTION -soThreshold 1000 -soPersistence ENABLE
show gslb vserver Vserver-GSLB-1
```

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:
 - o Methodâ€™ soMethod
 - o Thresholdâ€™ soThreshold
 - o Persistence Time-out (min) â€™ soPersistenceTimeout
3. Select the Persistence option and click **OK**.

Managing Client Connections

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.

This document includes the following information:

- [Enabling Delayed Cleanup of Virtual Server Connections](#)
- [Managing Local DNS Traffic by Using DNS Policies](#)
- [Adding DNS Views](#)

Enabling Delayed Cleanup of Virtual Server Connections

Updated: 2014-11-24

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>`

Example

```
> set gslb service Service-GSLB-1 -downStateFlush ENABLED
Done
> show gslb service Service-GSLB-1
Done
```

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

Updated: 2015-05-22

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.

This document includes the following information:

- [DNS Expressions](#)
- [Configuring DNS Actions](#)
- [Configuring DNS Policies](#)
- [Binding DNS Policies](#)

DNS Expressions

Updated: 2013-07-18

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

Updated: 2014-11-24

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>]`

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
Done
> show dns action dns_act_response_rewrite
1)      ActionName:  dns_act_response_rewrite ActionType:  Rewrite_Response TTL:  3600
Done
```

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:  vie
Done
```

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.nsl.*.* NA.tx
Done
> show dns action send_preferred_location
1)      ActionName:  send_preferred_location ActionType:  GslbPrefLoc PreferredLocList:  "
Done
```

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:
 - o Action Name (cannot be changed for an existing DNS action)
 - o 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.
 - o TTL (applicable only to the Rewrite Response action type)

Configuring DNS Policies

Updated: 2014-11-24

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 (â€™) 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:

- o 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.
- o 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.
- o 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.

- o A maximum of 64 policies are supported.
- o DNS policies are global to the NetScaler and cannot be applied to a specific virtual server or domain.
- o 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):

- o A
- o AAAA
- o NS
- o SRV
- o PTR
- o CNAME
- o SOA
- o MX
- o ANY

The following response codes (RCODE values) are also supported:

- o NOERROR - No error
- o FORMERR - Format error
- o SERVFAIL - Server failure
- o NXDOMAIN - Non-existent domain
- o NOTIMP - Query type not implemented
- o 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:

- o `add dns policy <name> <rule> <actionName>`
- o `show dns policy <name>`

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

To remove a configured DNS policy by using the command line interface

At the command prompt, type:

```
rm dns policy <name>
```

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:
 - o Policy Name (cannot be changed for an existing policy)
 - o Action
 - o 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 first 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

Updated: 2013-08-29

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:

- o bind dns global <policyName> <priority> [-gotoPriorityExpression <string>] [-type <type>]
- o show dns global -type <type>

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
```

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
```

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

Updated: 2014-11-24

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 [Configuring DNS Policies](#) and [Binding 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>

Example

```
add dns view PrivateSubnet
show dns view PrivateSubnet
```

To remove a DNS view by using the command line interface

At the command prompt, type:

```
rm dns view <viewName>
```

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, see [Configuring DNS Policies](#) and for details on how to bind DNS policies globally, see [Binding DNS Policies](#).

Configuring GSLB for Disaster Recovery

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.

This document includes the following information:

- [Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup](#)
- [Configuring for Disaster Recovery in an Active-Active Data Center Setup](#)
- [Configuring for Disaster Recovery with Weighted Round Robin](#)
- [Configuring for Disaster Recovery with Data Center Persistence](#)

Configuring GSLB for Disaster Recovery in an Active-Standby Data Center Setup

Updated: 2014-11-24

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>
```

Example

```
set gslb vserver vserver-GSLB-1 -backupVServer vserver-GSLB-2
```

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 GSLB a virtual server and GSLB services and bind the services to the virtual servers. 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

Updated: 2014-11-24

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 servers. 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:

- o `add lb vserver <name>@ -weight <WeightValue> <ServiceName>`
- o `set lb vserver <name>@ -weight <WeightValue> <ServiceName>`

Example

```
add lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
set lb vserver Vserver-LB-1 -weight 4 Service-HTTP-1
```

To set dynamic weight by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -dynamicWeight DynamicWeightType
```

Example

```
set gslb vserver Vserver-GSLB-1 -dynamicWeight ServiceWeight
```

To add weights to the GSLB services by using the command line interface

At the command prompt, type:

```
set gslb vserver <name> -serviceName GSLBServiceName -weight WeightValue
```

Example

```
set gslb vserver Vserver-GSLB-1 -serviceName Service-GSLB-1 -weight 1
```

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

Updated: 2014-11-24

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>

Example

```
set gslb service Service-GSLB-1 -sitePersistence HTTPRedirect -sitePrefix vserver-GSLB-1
show gslb service Service-GSLB-1
```

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).

Configuring GSLB for Proximity

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](#).

Configuring Parent-Child Topology

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.

There are three fundamental entities that must be configured for GSLB:

- **Site:** A GSLB site represents a NetScaler or a high availability (HA) pair of NetScaler appliances that maintain GSLB state information and provide information about how the NetScaler nodes should communicate. A site can also represent a data center.
- **GSLB virtual server:** A GSLB virtual server represents a group of resources to which users can be directed, and the logic used to select one resource versus another.
- **GSLB service:** A GSLB service represents a target resource and is bound to a GSLB virtual server. The target resource might be a load balancing virtual server on a NetScaler, or it could represent a third party server.

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.

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

1. To clearly separate GSLB sites. For example, to separate sites that participate in resolving DNS queries from the traffic management sites.
2. 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. Parent-child relationships can be used to build a two-level hierarchical GSLB design with the following characteristics:

- At the top level are parent sites that have peer relationships with other parents.
- Each parent can have multiple children, but each child can have only one parent.
- Each parent site exchanges health information with its children and with other parent sites.
- A child communicates only with its parent.

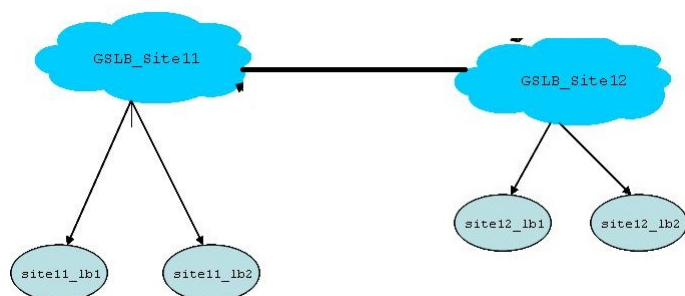
Note: In a parent-child relationship for GSLB, only the parent site does the GSLB resolution. The child sites act as normal load balancing sites.

Limitations of GSLB Parent-Child site configuration:

- You can configure 32 Parent sites and 1024 Child sites for each Parent site.
- On the Child site, by default, the `nwmetricExchange` and `sessionExchange` options are disabled.
- Round Trip Time (RTT) GSLB method is not recommended for GSLB Parent-Child site configuration.
- ADNS service or DNS load balancing virtual servers should be configured only in the Parent site.

Setting Up a Parent-Child Configuration for Global Server Load Balancing

If you have a firewall configured at a GSLB site, make sure that port 3011 is open. Follow the procedures at the following location to create services and virtual servers: [Configuring Global Server Load Balancing \(GSLB\)](#)
Figure 1. GSLB Parent-Child Topology



In the above figure:

- GSLB_Site11 and GSLB_Site12 are parent sites in a peer relationship.
- site11_lb1 and site11_lb2 are the child sites of GSLB_Site11, while site12_lb1 and site12_lb2 are the child sites of GSLB_Site12.

The configuration of each parent site includes the information about all the child sites associated with it, but the configuration of each child site pertains only to that child and its parent. A child site is not aware about any other parent site or other child sites in the configuration. For example, in the above figure, the configuration of child site site11_lb1 would include only information about its parent site, GSLB_Site11.

Note: GSLB auto sync syncs only the GSLB configuration across the parent sites. It does not sync any configuration to the child sites.

To set up a parent-child configuration for GSLB by using the NetScaler command line

1. On each parent site, enter the following command: `add gslb site<siteName><siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>] [-parentSite<string>]` For example: `# add gslb site gslb_site11 1.1.1.1 -publicIP 1.1.1.1`

```
# add gslb site site11_lb1 1.1.1.2 -publicIP 1.1.1.2 -parentSite gslb_site11
```

```
# add gslb site site11_lb2 1.1.1.3 -publicIP 1.1.1.3 -parentSite gslb_site11
```

```
# add gslb site gslb_site12 3.3.3.1 -publicIP 3.3.3.1
```

```
# add gslb site site12_lb1 3.3.3.2 -publicIP 3.3.3.2 -parentSite gslb_site12
```

```
# add gslb site site12_lb2 3.3.3.3 -publicIP 3.3.3.3 -parentSite gslb_site12
```

The above command makes the parent site aware of its child sites as well as of the other parent site in the configuration.

2. On each child site, enter the following command: `add gslb site<siteName><siteIPAddress> [-publicIP <ip_addr|ipv6_addr|*>] [-parentSite<string>]` For example: `# add gslb site site11_lb1 1.1.1.1 -publicIP 1.1.1.1`

```
# add gslb site site11_lb1 1.1.1.2 -publicIP 1.1.1.2 -parentSite gslb_site11
```

The above command creates the child site and adds the parent-site information to child site's configuration.

Network metrics, such as RTT and persistence session information, are synced only across the parent sites. Therefore, parameters like `nwMetric` 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.

Note: If you want to use different private and public IP address for GSLB services, add the corresponding GSLB-service related configuration to the child site in a separate procedure, not as part of the GSLB site configuration.

Link Load Balancing

Link load balancing (LLB) balances outbound traffic across multiple Internet connections provided by different service providers. LLB enables the Citrix® 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 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.

This document includes the following information:

- [Configuring a Basic LLB Setup](#)
- [Configuring RNAT with LLB](#)
- [Configuring a Backup Route](#)
- [Resilient LLB Deployment Scenario](#)
- [Monitoring an LLB Setup](#)

Configuring a Basic LLB Setup

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](#)

Configuring Services

Updated: 2014-10-27

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>`

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
```

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:
 - o Service Name*â€”name
 - o Serverâ€”IP
 - o Protocol*â€”serviceType (Select ANY from the drop-down list.)
 - o Port*â€”port

* A required parameter
4. Click Create.
5. Repeat Steps 2-4 to create another service.
6. Click Close.
7. 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.

Configuring an LLB Virtual Server and Binding a Service

Updated: 2014-10-28

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:

- o add lb vserver <name> <serviceType>
- o bind lb vserver < name> <serviceName>
- o show lb vserver < name>

Example

```
add lb vserver Router1-vip any
bind lb vserver Router-vip ISP1R_svc_any
sh lb vserver router-vip
Router-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
```

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.
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.

Configuring the LLB Method and Persistence

Updated: 2014-10-28

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:

- o set lb vserver <name> -lbMethod <lbMethod> -persistenceType <persistenceType>
- o show lb vserver <name>

Example

```
set lb vserver router-vip -lbmethod ROUNDROBIN -persistencetype SOURCEIP

show lb vserver Router-vip
Router-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 Tim
Connection Failover: DISABLED
```

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** section, select Method and configure the load balancing method.
3. In the **Advanced** section, select **Persistence** and configure the persistence parameters.

Configuring an LLB Route

Updated: 2014-10-28

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 virtual server as the gateway. A route is a collection of links that are load balanced. Requests are sent to the 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:

- o add lb route <network> <netmask> <gatewayName>
- o show lb route [<network> <netmask>]

Example

```
add lb route 0.0.0.0 0.0.0.0 Router-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	Router-vip	UP

To configure an IPv6 LLB route by using the command line interface

At the command prompt, type:

- o add lb route6 <network> <gatewayName>
- o show lb route6

```
add lb route6 ::/0 llb6_vs
show lb route6
```

	Network	VIP	Flags
1)	::/0	llb6_vs	UP

Example

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 IPv6 route.
3. In the Create LB Route or Create LB IPV6 Routedialog box, set the following parameters:

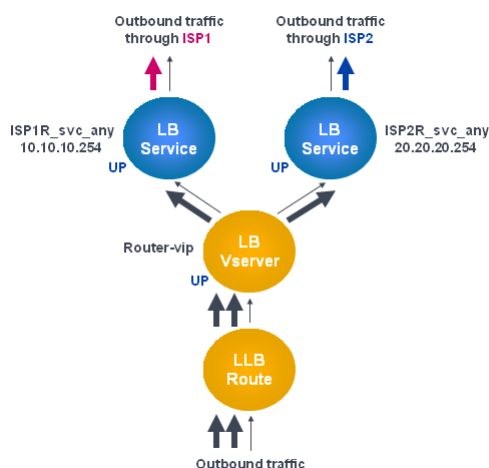
- Network*
- Netmask*â€”Required for IPV4 routes.
- Gateway Name*â€”gatewayName

* A required parameter

4. 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



Note: If your Internet service provider has provided an IPv6 address, replace the IPv4 service with an IPv6 service in the above figure.

Creating and Binding a Transparent Monitor

Updated: 2014-10-28

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:

- o add lb monitor <monitorName> <type> -destIP <ip_addr|*> -transparent YES
- o show lb monitor [<monitorName>]

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
```

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:

- o Name*
- o Type*
- o Destination IP
- o Transparent

* A required parameter

4. Click Create, and then click Close.
5. 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:

- o bind lb monitor <monitorName> <serviceName>
- o show service <name>

Example

```
bind lb monitor monitor-HTTP-1 isPlR_svc_any
Done
> show service isPlR_svc_any
ISPlR_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

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

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.

To configure RNAT by using the command line interface

At the command prompt, type:

- o set rnat <network> <netmask>
- o show rnat

Example

```
set rnat 10.102.29.0 255.255.255.0
> show rnat
1)      Network: 10.102.29.0      Netmask: 255.255.255.0
      NatIP: *
```

To configure RNAT by using the configuration utility

1. Navigate to System > Network > Routes.
2. On the **RNAT** tab, from the **Actions** drop-down list, select **Configure RNAT**.
3. Specify the network on which to perform RNAT.

To enable Use Subnet IP mode by using the command line interface

At the command prompt, type:

- o enable ns mode USNIP
- o show ns mode

Example

```
enable ns mode USNIP
> show ns mode
```

	Mode	Acronym	Status
	-----	-----	-----
1)	Fast Ramp	FR	ON
2)	â€¦		
8)	Use Subnet IP	USNIP	ON
9)	â€¦		

To enable Use Subnet IP mode by using the configuration utility

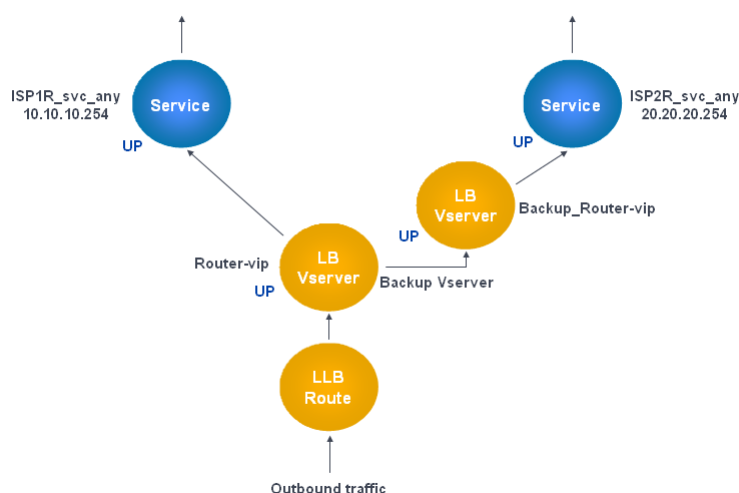
1. Navigate to System > Settings and, under **Modes and Features**, click **Configure Modes**.
2. In the **Configure Modes** dialog box, select **Use Subnet IP**, and then click **OK**.

Configuring a Backup Route

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.

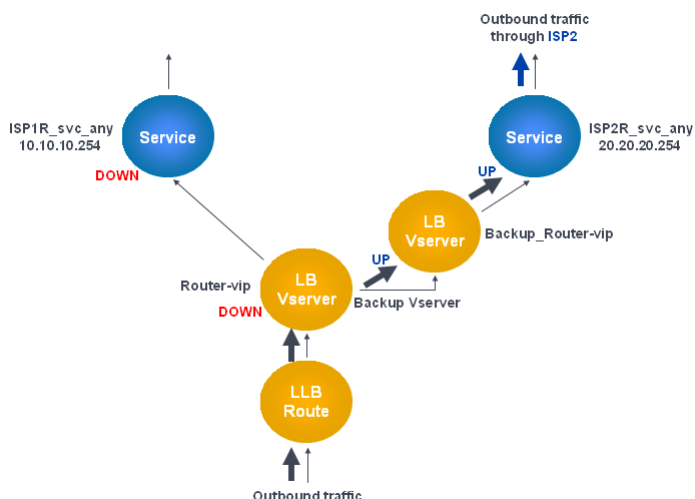
Figure 1. Backup Route Setup



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



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>
```

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 v6MaskLe
Backup: Router2-vip
Connection Failover: DISABLED
Done

```

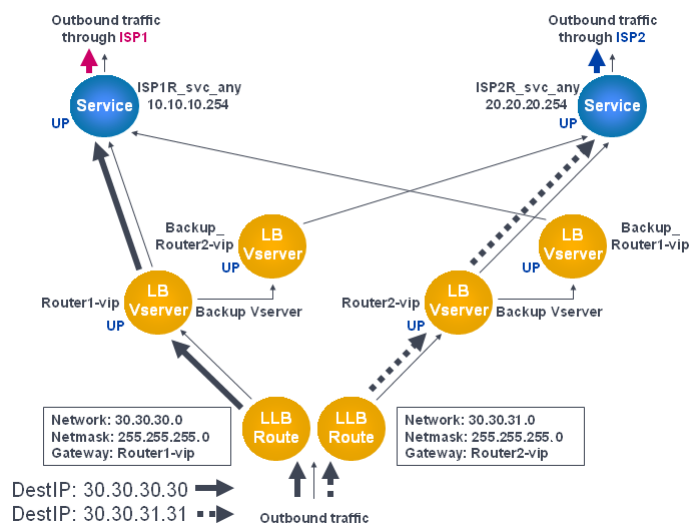
To set the secondary virtual server as the backup virtual server by using the configuration utility

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

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**.

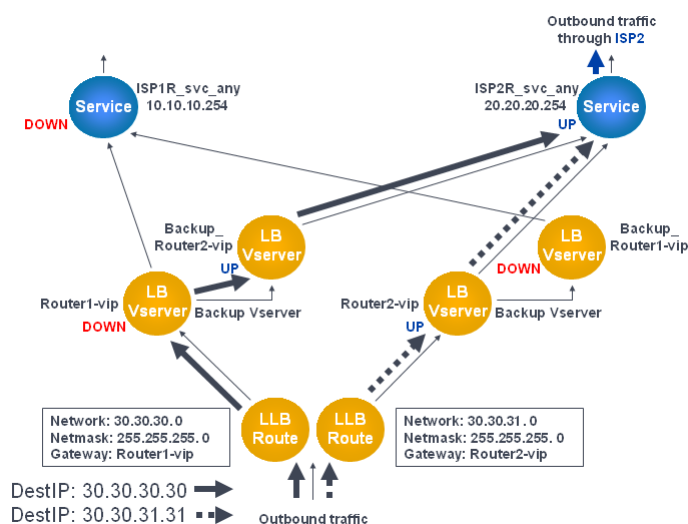
Figure 1. Resilient LLB Deployment Setup



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



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

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

Updated: 2013-09-05

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>]
```

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
BRVSERVER	10.10.1.1	80	HTTP	DOWN	0/s	0/s
LBVIP	10.102.29.66	80	HTTP	UP	0/s	0/s
Done						

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>
```

Example

```
stat service Service-HTTP-1
```

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

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.

The following video explains a basic load balancing configuration.

How Load Balancing Works

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](#).

To understand how load balancing works, see the following sections:

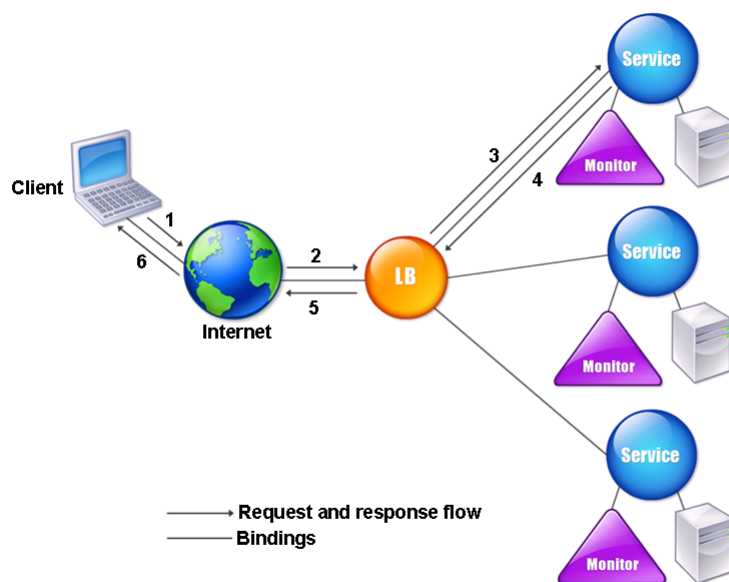
- [Load Balancing Basics](#)
- [Understanding the Topology](#)
- [Use of Wildcards Instead of IP Addresses and Ports](#)
- [Configuring Global HTTP Ports](#)

Load Balancing Basics

Updated: 2013-06-18

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.

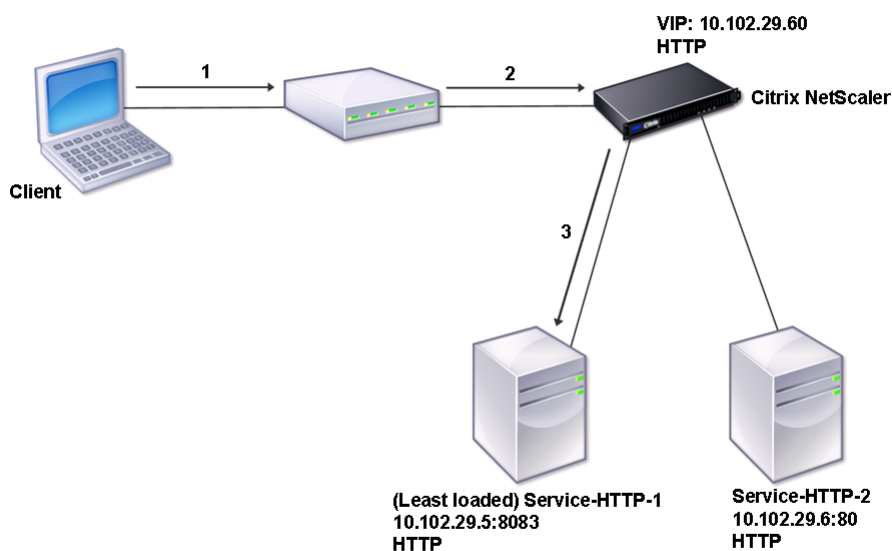
For variations in the load balancing setup, see the following use cases:

- [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 Balancing RDP services](#)

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



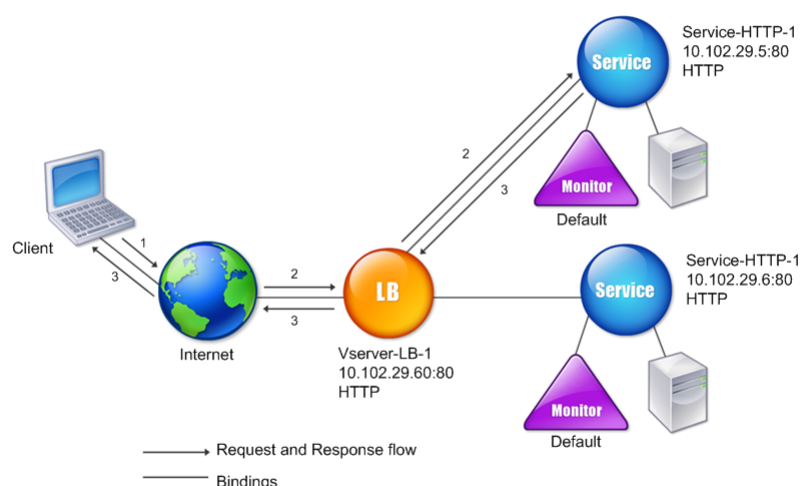
In the diagram, load balancing is used to manage traffic flow to the servers. The virtual server selects the service and assigns it to serve client requests. Consider a scenario where the services Service-HTTP-1 and Service-HTTP-2 are created and bound to the virtual server named Vserver-LB-1. Vserver-LB-1 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	Mandatory Parameters and Sample Values			
	Name	IP Address	Port	Protocol
Virtual server	Vserver-LB-1	10.102.29.60	80	HTTP
Services	Service-HTTP-1	10.102.29.5	8083	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



Use of Wildcards Instead of IP Addresses and Ports

Updated: 2013-11-20

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 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, <code>set ns param - httpPort 80</code>). 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.
*	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 <code>clearText</code> 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

Updated: 2013-10-23

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:

- o set ns param -httpPort <port>
- o show ns param

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
...
...
```

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
>
```

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.

Setting Up Basic Load Balancing

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.

To set up basic load balancing, see the following sections:

- [Enabling Load Balancing](#)
- [Configuring Services](#)
- [Creating a Virtual Server](#)
- [Binding Services to the Virtual Server](#)
- [Verifying the Configuration](#)

Enabling Load Balancing

Updated: 2015-05-21

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
```

To enable load balancing by using the configuration utility

Navigate to System > Settings and, in Configure Basic Features, select Load Balancing.

Creating a Virtual Server

Updated: 2013-09-02

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 Web site, application, or server through the virtual server's IP address or FQDN.

Note: The virtual server is designated as DOWN until you bind the services that you created to it, and until the NetScaler 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 command line interface

At the command prompt, type:

```
add lb vserver <name> <serviceType> <ip> <port>
```

Example

```
add lb vserver Vserver-LB-1 HTTP 10.102.29.60 80
```

To create a virtual server by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and then create a virtual server.

Binding Services to the Virtual Server

Updated: 2014-09-03

After you have created services and a virtual server, you must bind the services to the virtual server. In most cases, 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 command line interface

At the command prompt, type:

```
bind lb vserver <name> <serviceName>
```

Example

```
bind lb vserver Vserver-LB-1 Service-HTTP-1
```

To bind a service to a load balancing 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.

Load Balancing Algorithms

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.
SRCIPSRCPORHASH	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	Specifies
-------------	-----------

	Types of Load Balanced Service	
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 NetScaler switches to the specified load balancing method depending on the request rate.
- If the Startup RR Factor is any number other than zero, NetScaler 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 command line interface

At the command prompt, type:

```
set lb parameter -startupRRFactor <positive_integer>
```

Example

```
set lb parameter -startupRRFactor 25000
```

To set the startup round-robin factor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing Parameters, and set the Startup RR Factor.

The Least Connection Method

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:

- o **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.
- o **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.

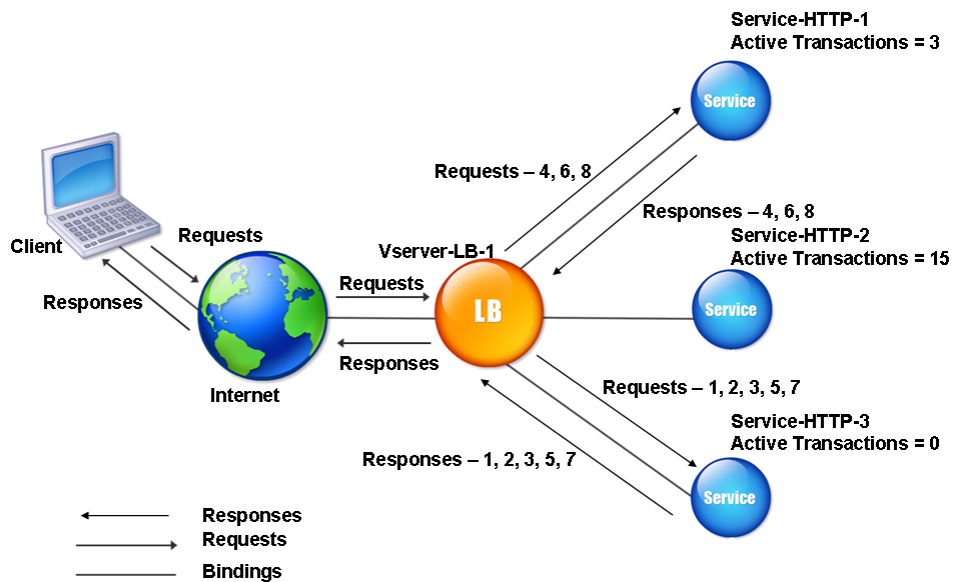
For Real-Time Streaming Protocol (RTSP) connections, the NetScaler appliance uses the number of active control connections to determine the lowest number of connections to an RTSP service.

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:

- o Service-HTTP-1 is handling 3 active transactions.
- o Service-HTTP-2 is handling 15 active transactions.
- o 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



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. Â
Request-2	Service-HTTP-3 (N = 1)	2	
Request-3	Service-HTTP-3	3	

	(N = 2)		
Request-4	Service-HTTP-1 (N = 3)	4	Service-HTTP-1 and Service-HTTP-3 have the same number of active connections.
Request-5	Service-HTTP-3 (N = 3)	4	
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:

$$Nw = (\text{Number of active transactions}) * (10000 / \text{weight})$$

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 NetScaler performs load balancing in a round robin manner. Therefore, Service-HTTP-3 receives the ninth request.

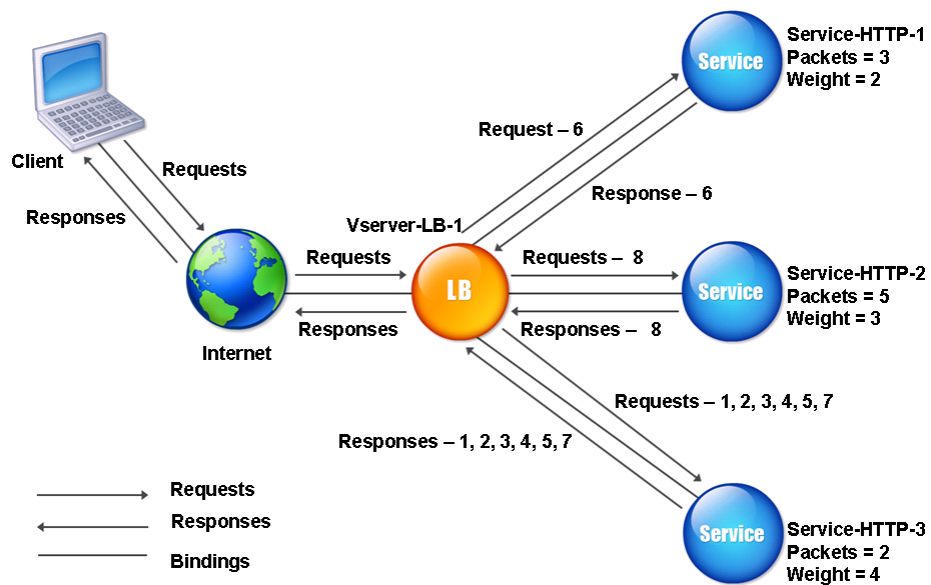
The following table explains how connections are distributed on the three-service load balancing setup that is described above.

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



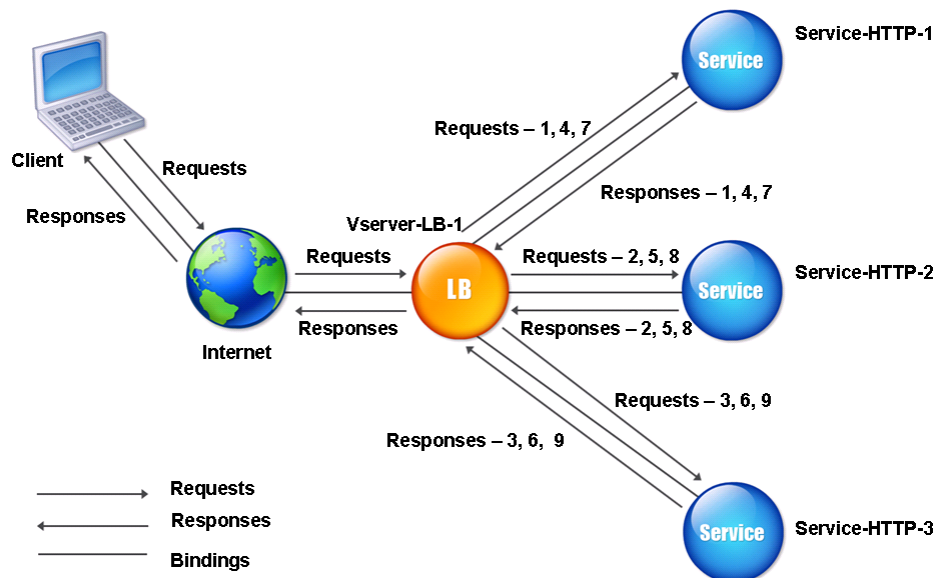
To configure the least connection method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

The Round Robin Method

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



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:

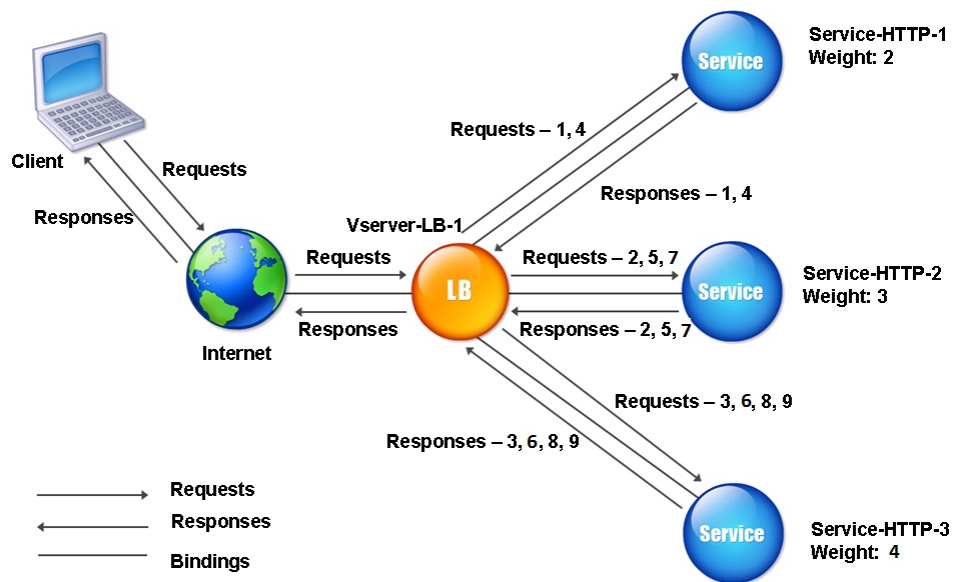
- Service-HTTP-1 receives the first request.
- Service-HTTP-2 receives the second request.
- Service-HTTP-3 receives the third request.
- Service-HTTP-1 receives the fourth request.
- Service-HTTP-2 receives the fifth request.
- Service-HTTP-3 receives the sixth request.
- Service-HTTP-2 receives the 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



To configure the round robin method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

The Least Response Time Method

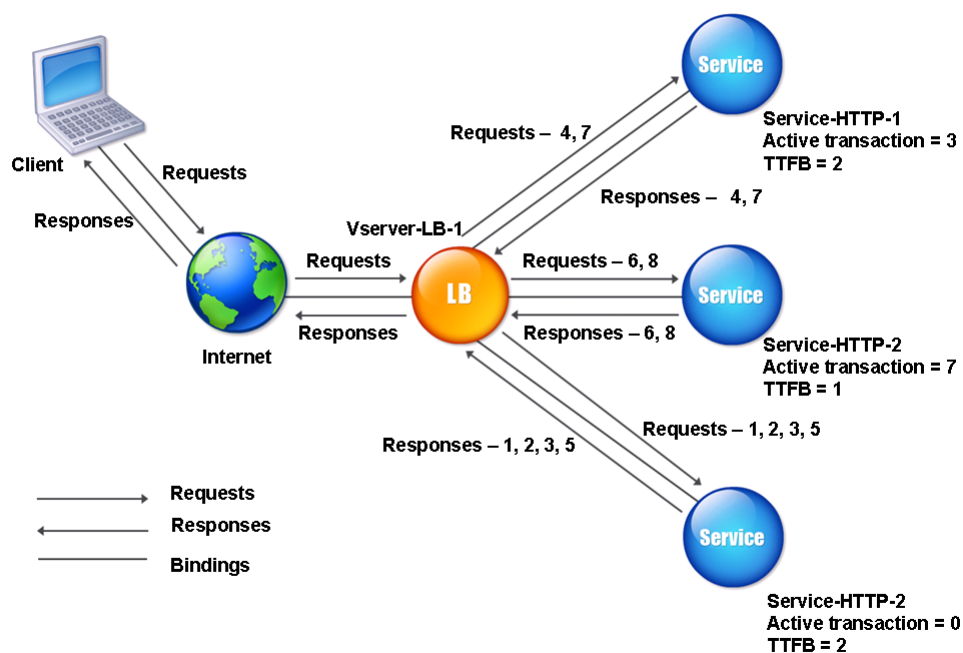
When the load balancing virtual server is configured to use the least response time method, it selects the service with the fewest active connections and the lowest average response time. You can configure this method for HTTP and Secure Sockets Layer (SSL) services only. The response time (also called 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



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 NetScaler 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
			Service-HTTP-3 has the lowest N value.

Request-1	Service-HTTP-3 (N = 0)	N = 2	
Request-2	Service-HTTP-3 (N = 2)	N = 4	Â
Request-3	Service-HTTP-3 (N = 3)	N = 6	Â
Request-4	Service-HTTP-1 (N = 6)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-5	Service-HTTP-3 (N = 6)	N = 8	
Request-6	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-2 has the lowest N value.
Request-7	Service-HTTP-1 (N = 8)	N = 15	Service-HTTP-1, Service-HTTP-2, and Service-HTTP-3 have the same N values.
Request-8	Service-HTTP-2 (N = 8)	N = 9	

The virtual server selects a service by using the value (Nw) in the following expression:

$$Nw = (N) * (10000 / \text{weight})$$

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 NetScaler 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.

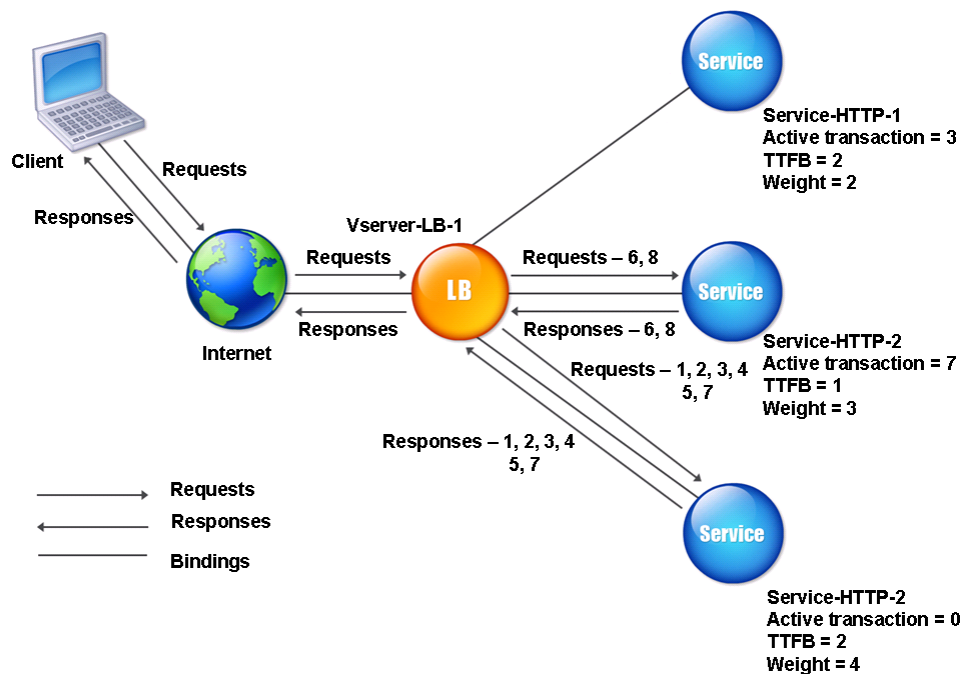
Service-HTTP-1 has the lowest weight and therefore the highest Nw value, so the virtual server does not select it for load balancing.

The following table explains how connections are distributed in the three-service load balancing setup described above.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	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 = 15000	^
Request-4	Service-HTTP-3 (Nw = 15000)	Nw = 20000	^
Request-5	Service-HTTP-3 (Nw = 20000)	Nw = 25000	^
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 = 33333.34	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 values of other services (Service-HTTP-2 and Service-HTTP-3) are equal to 105000.			

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



To configure the least response time method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

When a load balancing virtual server is configured to use the least response time method with monitors, it uses the existing monitoring infrastructure to select the service with the smallest number of active transactions and the fastest average response time. Before you use the least response time method with monitoring, you must bind application-specific monitors to each service and enable least response time method mode on these monitors. The NetScaler appliance then makes load balancing decisions based on the response times it calculates from monitoring probes. For more information about configuring monitors, see [Configuring Monitors in a Load Balancing Setup](#).

You can use the least response time method with monitors to select non-HTTP and non-HTTPS services. 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.
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.

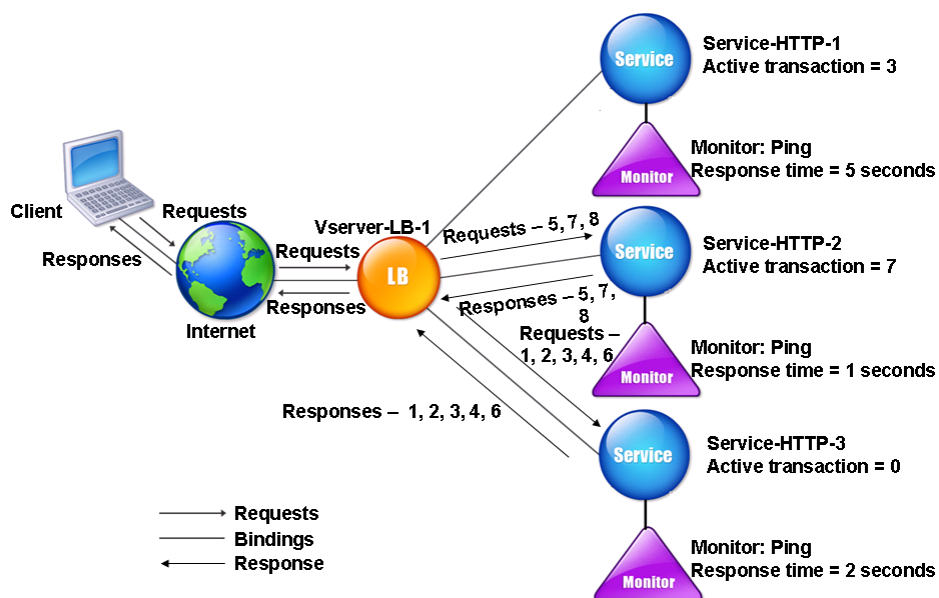
DNS	Time difference between a DNS query and the DNS response.
TCP	Time difference between a SYN request and the SSL handshake completion.
FTP	Time difference between the sending of the user name and the completion of user authentication.
HTTPS (monitors HTTPS requests)	Time difference is same as for 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 least response time method with monitors. 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 3. How the Least Response Time Load Balancing Method Works When Using Monitors



The virtual server selects a service by using the value (N) in the following expression:

$$N = \text{Number of active transactions} * \text{Response time that is determined by the monitor}$$

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.

- o 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.
- o 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)	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	^
Request-3	Service-HTTP-3 (N = 4)	N = 6	^
Request-4	Service-HTTP-3 (N = 6)	N = 8	^
Request-5	Service-HTTP-2 (N = 7)	N = 8	Service-HTTP-1 and Service-HTTP-3 have the same N values.
Request-6	Service-HTTP-3 (N = 8)	N = 10	
Request-7	Service-HTTP-2 (N = 8)	N = 9	Service-HTTP-2 has the lowest N value.
Request-8	Service-HTTP-1 (N = 9)	N = 10	^
Service-HTTP-1 is again selected for load balancing when it completes its active transactions or when the N value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 15.			

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:

$$Nw = (N) * (10000 / \text{weight})$$

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 request, because it is not handling any active transactions.
- Service-HTTP-3 receives the second, third, and fourth requests, because this service has the lowest Nw value.
- Service-HTTP-2 receives the fifth request, because this service has the lowest Nw value.
- Service-HTTP-3 receives the sixth request, because this service has the lowest Nw value.
- Service-HTTP-2 receives the seventh and 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.

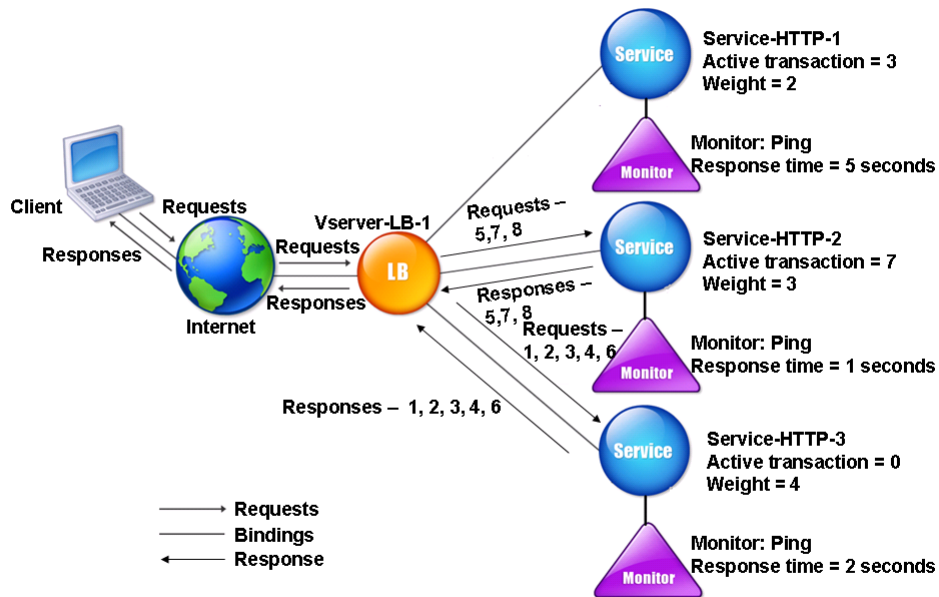
The following table summarizes how Nw is calculated for various monitors.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (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	^
Request-3	Service-HTTP-3 (Nw = 15000)	Nw = 20000	^
Request-4	Service-HTTP-3 (Nw = 20000)	Nw = 25000	^
Request-5	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-6	Service-HTTP-3 (Nw = 25000)	Nw = 30000	Service-HTTP-3 has the lowest Nw value.
Request-7	Service-HTTP-2 (Nw = 23333.34)	Nw = 26666.67	Service-HTTP-2 has the lowest Nw value.
Request-8	Service-HTTP-2	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 the Nw value of the other services (Service-HTTP-2 and Service-HTTP-3) is equal to 75000.

The following diagram illustrates how the virtual server uses the least response time method when weights are assigned.

Figure 4. How the Least Response Time Load Balancing Method with Monitors Works When Weights Are Assigned



To configure the least response time method using monitors, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

About Hashing Methods

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 provides the following hashing methods:

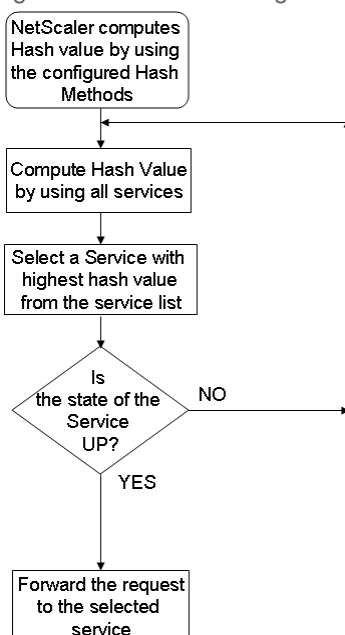
- URL hash method
- Domain hash method
- Destination IP hash method
- Source IP hash method
- Source IP Destination IP hash method
- Source IP Source Port hash method
- Call ID hash method
- Token method

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.

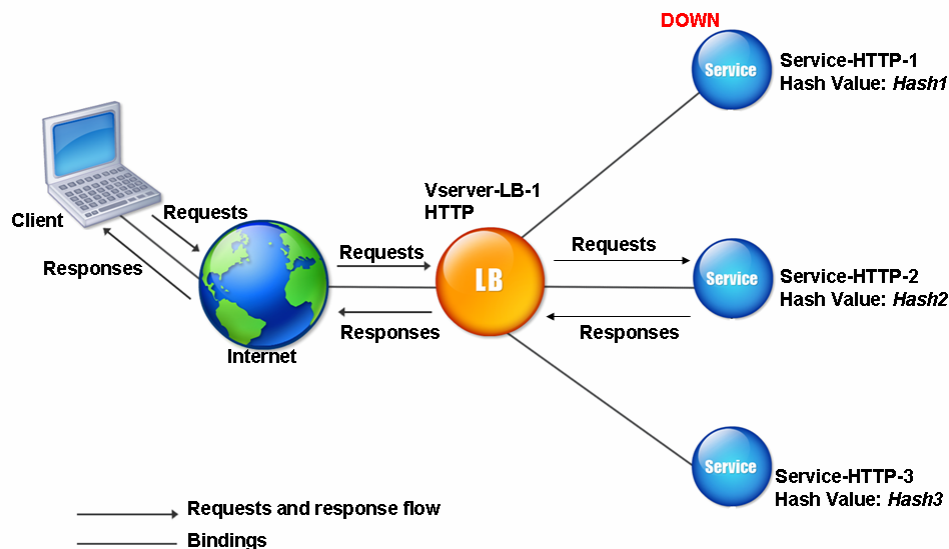
Figure 1. How the Hashing Methods Distribute Requests



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 NetScaler 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



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

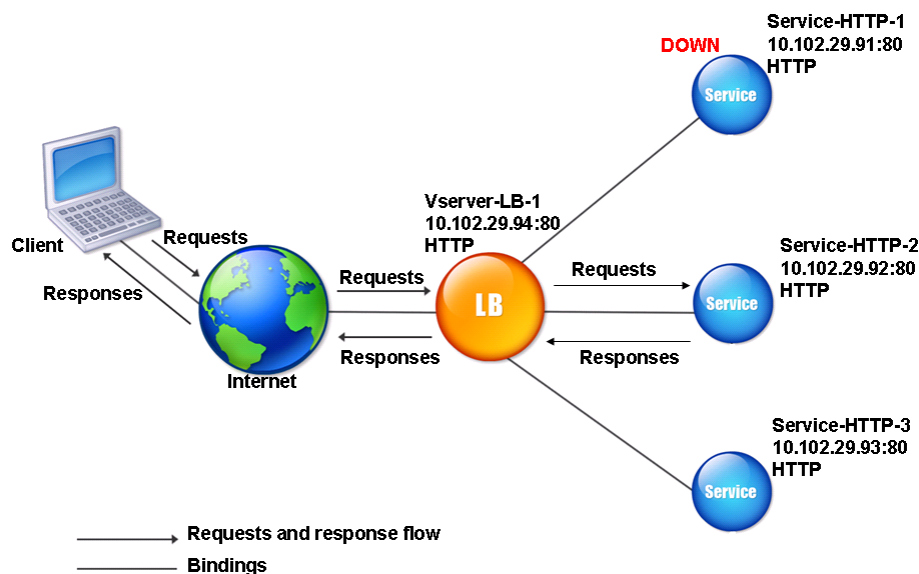
When you configure the NetScaler appliance to use the URL hash method for load balancing the services, for selecting a service, the NetScaler generates a hash value of the HTTP URL present in the incoming request. If the service selected by the hash value is DOWN, the algorithm has a method to select another service from the list of active services. The NetScaler 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 NetScaler 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, NetScaler uses a specific algorithm and considers a part of the URL. By default, the NetScaler 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. NetScaler selects Service-HTTP-1. If Service-HTTP-1 is DOWN, the NetScaler selects Service-HTTP-2.

The following diagram illustrates this process.

Figure 3. How URL Hashing Operates



If both Service-HTTP-1 and Service-HTTP-2 are DOWN, NetScaler 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 NetScaler 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

Updated: 2013-09-03

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 [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](#).

The Least Bandwidth Method

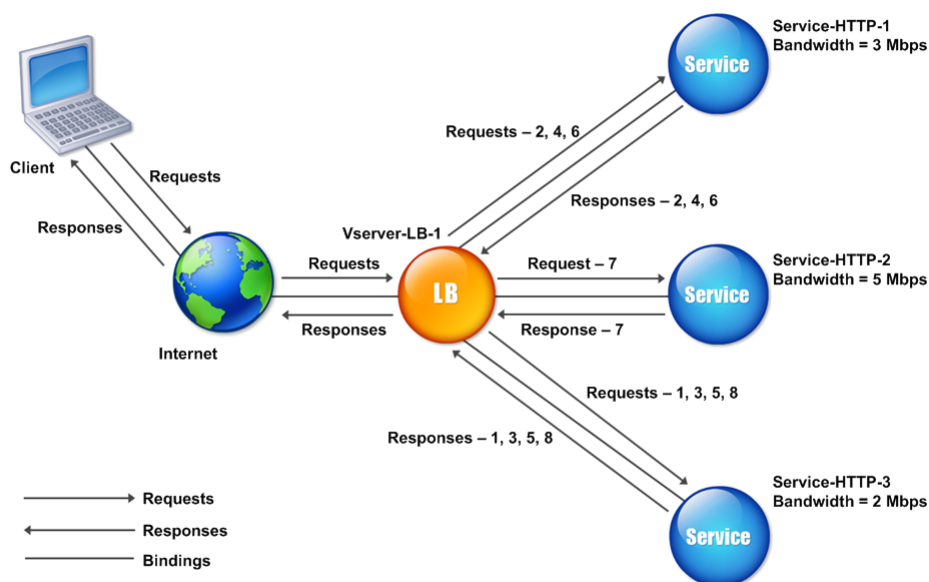
A load balancing virtual server configured to use the least bandwidth method selects the service that is currently serving the least amount of traffic, measured in megabits per second (Mbps). The following example shows how the virtual server selects a service for load balancing by using the least bandwidth 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



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-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	
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	
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:

$$Nw = (N) * (10000 / \text{weight})$$

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.

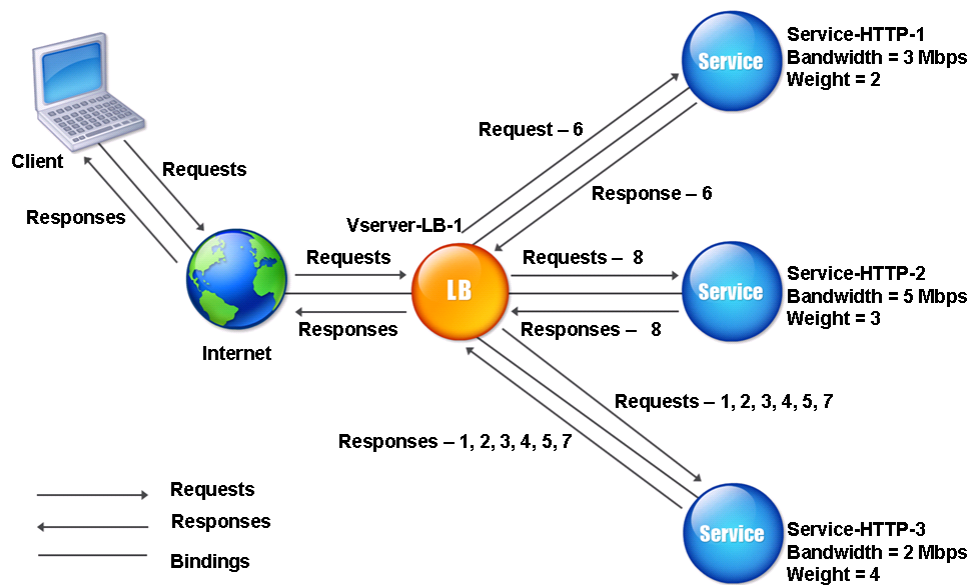
The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-3	Nw = 5000	Service-HTTP-3 has the lowest Nw value.

	(Nw = 5000)		
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	Â
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	Â
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	Â
Request-5	Service-HTTP-3 (Nw = 12500)	Nw = 15000	Â
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	
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



To configure the least bandwidth method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

The Least Packets Method

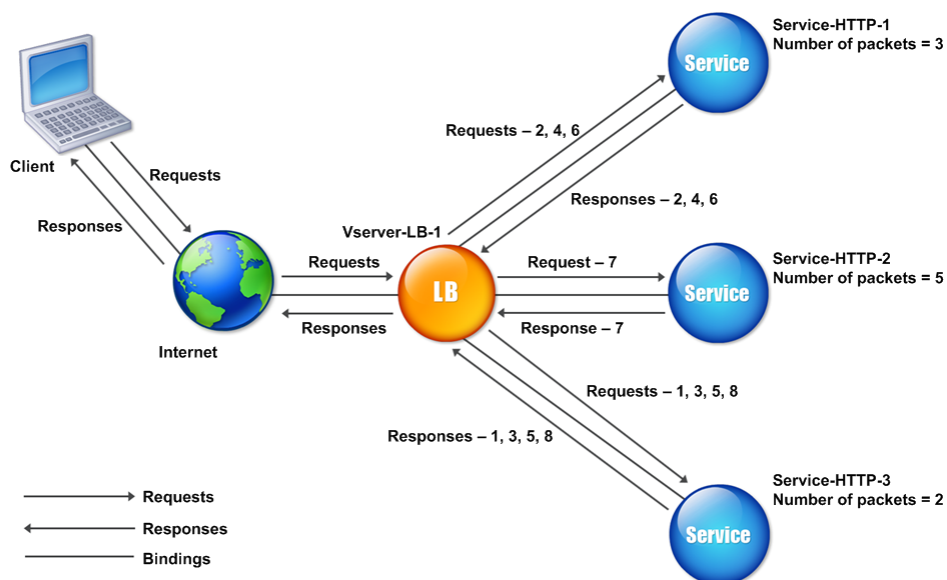
A load balancing virtual server configured to use the least packets method selects the service that has received the fewest packets in the last 14 seconds.

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



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-2		N = 4	Service-HTTP-1 and Service-HTTP-3 have the same N values.

	Service-HTTP-1 (N = 3)		
Request-3	Service-HTTP-3 (N = 3)	N = 4	
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	
Request-8	Service-HTTP-3 (N = 5)	N = 6	^

Note: If you enable the RTSP NAT option on the virtual server, the NetScaler 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:

$$Nw = (N) * (10000 / \text{weight})$$

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.

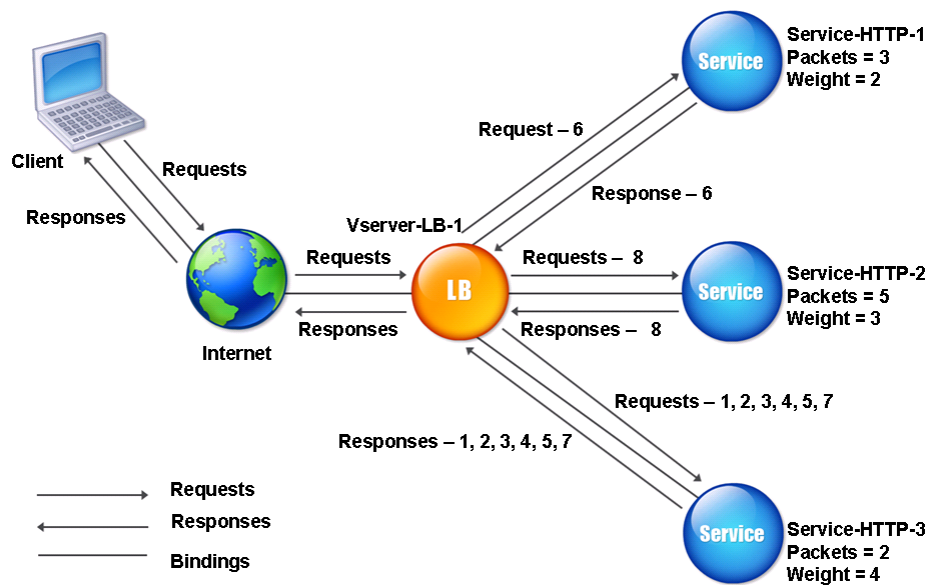
The following table summarizes how Nw is calculated.

Request Received	Service Selected	Current Nw Value (Number of Active Transactions) * (10000 / weight)	Remarks
Request-1	Service-HTTP-3	Nw = 5000	Service-HTTP-3 has the lowest Nw value.

	(Nw = 5000)		
Request-2	Service-HTTP-3 (Nw = 5000)	Nw = 7500	Â
Request-3	Service-HTTP-3 (Nw = 7500)	Nw = 10000	Â
Request-4	Service-HTTP-3 (Nw = 10000)	Nw = 12500	Â
Request-5	Service-HTTP-3 (Nw = 12500)	Nw = 15000	Â
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	
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



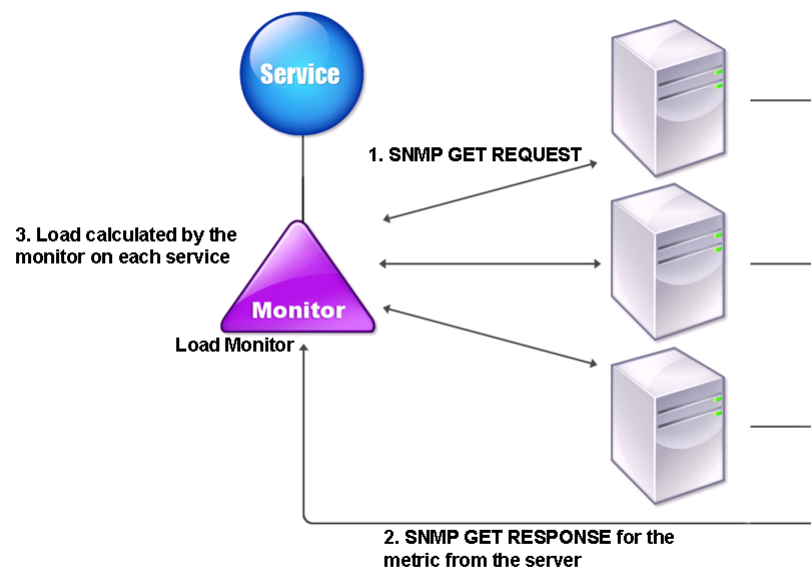
To configure the least packets method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

The Custom Load Method

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



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:

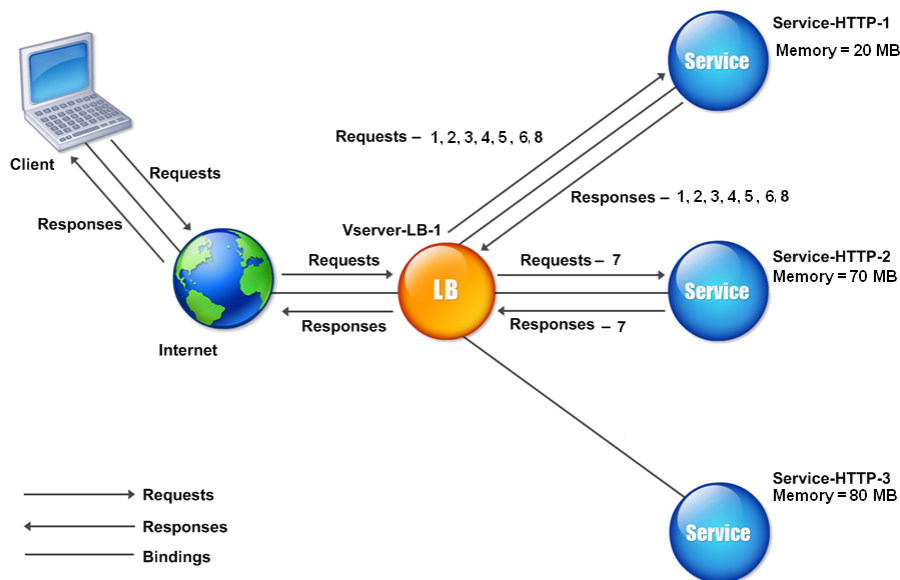
- Metrics values retrieved through SNMP probes that exist as tables in the NetScaler.
- Threshold value set for each metric.
- Weight assigned to each metric.

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



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	^
		N = 70	^

Request-5	Service-HTTP-1 (N = 60)		
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	
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:

$$Nw = (N) * (10000 / \text{weight})$$

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.

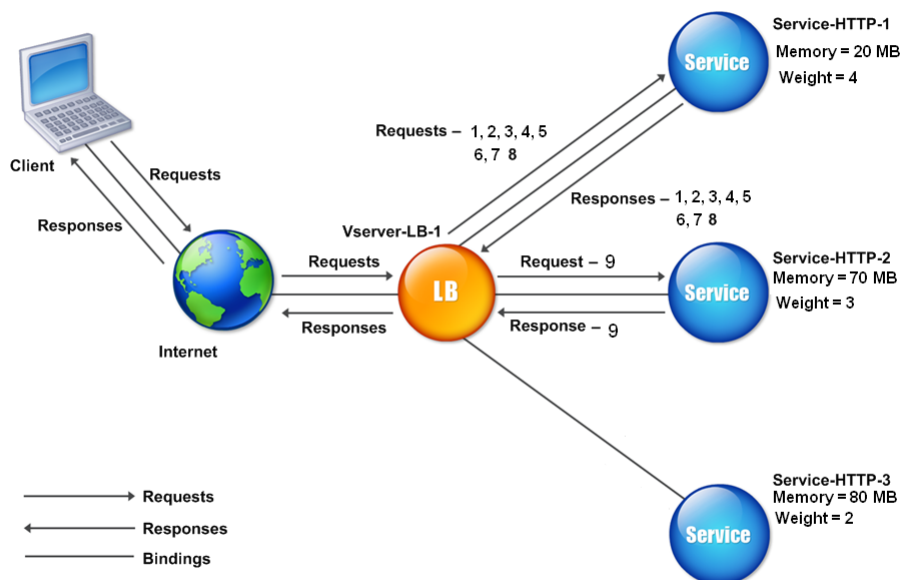
Request received	Service selected	Current Nw Value (Number of Active Transactions) * (10000 / Weight)	Remarks
Request-1	Service-HTTP-1 (Nw = 50000)	Nw = 75000	Service-HTTP-1 has the lowest Nw value.
Request-2	Service-HTTP-1 (Nw = 5000)	Nw = 100000	Â
Request-3	Service-HTTP-1 (Nw = 15000)	Nw = 125000	Â
Request-4	Service-HTTP-1 (Nw = 20000)	Nw = 150000	Â
Request-5		Nw = 175000	Â

	Service-HTTP-1 (Nw = 23333.34))		
Request-6	Service-HTTP-1 (Nw = 25000)	Nw = 200000	Â
Request-7	Service-HTTP-1 (Nw = 23333.34)	Nw = 225000	Â
Request-8	Service-HTTP-1 (Nw = 25000)	Nw = 250000	Â
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



To configure the custom load method, see [Configuring a Load Balancing Method that Does Not Include a Policy](#).

Configuring the Token Method

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.

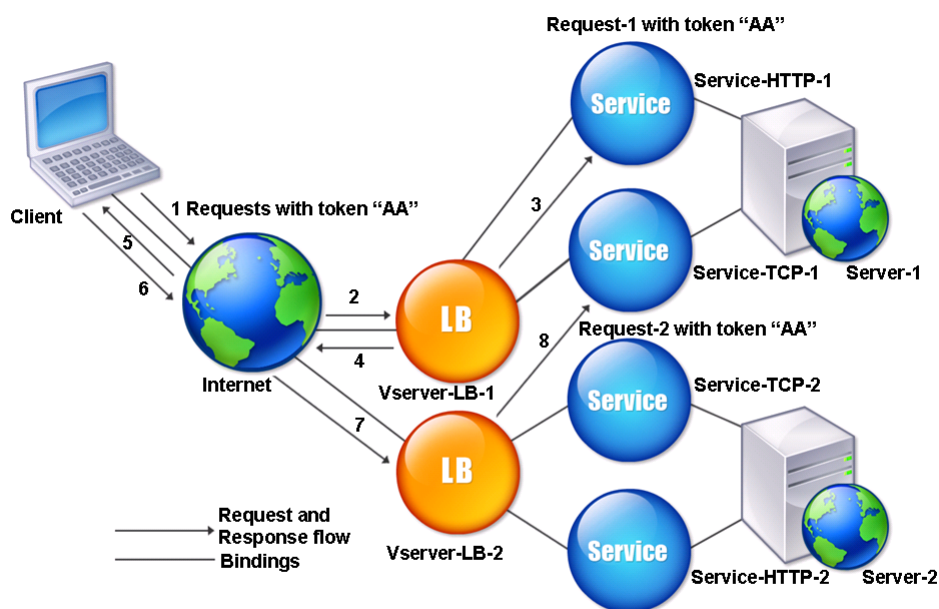
This method is content aware; it operates differently for TCP, HTTP, and HTTPS connections. For HTTP or HTTPS services, the token is found in the HTTP headers, the URL, or the BODY. To locate the token, you specify or create a classic or advanced expression. For more information on classic or advanced 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 NetScaler 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



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:

- o set lb vserver <name> -lbMethod TOKEN -rule <rule> -datalength <length> -dataoffset <offset>
- o show lb vserver <name>

Example

```
set lb vserver LB-VServer-1 -lbMethod TOKEN -rule 'AA' -datalength 2 -dataoffset 25  
show lb vserver LB-VServer-1
```

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.

Configuring a Load Balancing Method That Does Not Include a Policy

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 NetScaler command line 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 [Configuring the 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 [The Destination IP Hash Method](#), [The Source IP Hash Method](#), [The Source IP Destination IP Hash Method](#), and [The Source IP Source Port Hash Method](#).

To set the load balancing method by using the command line interface

At the command prompt, type:

```
set lb vserver <name> -lbMethod <method>
```

Example

```
set lb vserver Vserver-LB-1 -lbMethod LeastConnection
```

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

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

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.

Table 1. Types of Persistence

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.
User-Defined Rule	RULE. Connections that match a user-defined rule are treated as parts of the same persistence session.

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.

Table 2. Persistence Types and Numbers of Simultaneous Connections Supported

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.

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.

Table 3. Relationship of Persistence Type to Virtual Server Type

Persistence Type	HTTP	HTTPS	TCP	UDP/IP	SSL_Bridge	SSL_TCP	RTSP
SOURCEIP	YES	YES	YES	YES	YES	YES	NO
COOKIEINSERT	YES	YES	NO	NO	NO	NO	NO
SSLSESSION	NO	YES	NO	NO	YES	YES	NO
URLPASSIVE	YES	YES	NO	NO	NO	NO	NO
CUSTOMSERVERID	YES	YES	NO	NO	NO	NO	NO
RULE	YES	YES	YES	NO	NO		NO
SRCIPDESTIP	YES	YES	YES	YES	YES	YES	NO
DESTIP	YES	YES	YES	YES	YES	YES	NO
CALLID	NO	NO	NO	NO	NO	NO	NO
RTSPID	NO	NO	NO	NO	NO	NO	YES

Persistence Based on Source IP Address

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.

Persistence Based on HTTP Cookies

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 NetScaler 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 nonscriptable 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)`

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
```

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 -useSecuredPersistenceCookie ENABLED -cookiePassphrase test
```

To encrypt the cookie by using the configuration utility, navigate to Traffic Management > Change Load Balancing Parameters, and select Use Secured Persistence Cookie and Cookie Passphrase and enter a passphrase.

Persistence Based on SSL Session IDs

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, the NetScaler appliance does not encrypt or decrypt data when it forwards requests to services in an SSL bridge configuration, because it must maintain the data structures to keep track of the sessions. This type of persistence therefore 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 Web site being interrupted and the client being required to reauthenticate or restart the session. It may also result in large numbers 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](#).

Persistence Based on Diameter AVP Number

You can use persistence based on the 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>
```

Example

```
set lb vserver diameter_vs -persistenceType DIAMETER -persistAVPno 263
```


Custom Server ID Persistence

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 -serverID 2345-photo-56789
set service Service-2 10.102.29.6 -serverID 2345-drawing-abb123
```

On virtual server Web11, enable Custom Server ID persistence.

Example

```
set lb vserver Web11 -persistenceType customserverID
bind lb vserver Web11 Service-[1-2]
```

Create the following expression so that all URL queries containing the string "sid=" are examined.

```
HTTP.REQ.URL.AFTER_STR(â€œsid=â€œ)
```

When a client sends a request with the following URL to the IP address of Web11, the NetScaler 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.

Persistence Based on IP Addresses

You can base persistence on Destination IP addresses, or on both Source IP and Destination IP Addresses.

Persistence Based on Destination IP Addresses

Updated: 2013-09-24

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](#).

Persistence Based on SIP Call ID

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](#).

Persistence Based on RTSP Session IDs

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.

Configuring URL Passive Persistence

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 [Policy Configuration and Reference](#).

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(â€œurlp=â€•)
```

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 NetScaler 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 command line interface

At the command prompt, type:

```
set lb vserver <vserverName> [-rule <expression>]
```

Example

```
set lb vserver LB-VServer-1 â€"rule HTTP.REQ.URL.AFTER_STR(â€œurlp=â€•)
```

To configure URL passive persistence 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 URLPASSIVE.

Configuring Persistence Based on User-Defined Rules

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 [Policy Configuration and Reference](#).

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 NetScaler command line.

To configure persistence based on user-defined rules by using the command line interface

At the command prompt, type:

```
set lb vserver <vserverName> [-rule <expression>][-resRule <expression>]
```

Example

```
set lb vserver vsvr_name â€"rule http.req.header("cookie").value(0).typecast_nvlist_t('=' ,';' ;
set lb vserver vsvr_name â€"resrule http.res.header("set-cookie").value(0).typecast_nvlist_t
```

To configure persistence based on user-defined rules 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, 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")
```

Configuring Persistence Types That Do Not Require a Rule

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 command line interface

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`

Example

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP
```

```
show lb vserver
```

Note: For IP-based persistence, you can also set the `persistMask` parameter.

To configure persistence on 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 persistence type other than RULE.

Configuring Backup Persistence

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 HTTP 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>
```

Example

```
set lb vserver Vserver-LB-1 -persistenceType CookieInsert -persistenceBackup SourceIP
```

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.

Configuring Persistence Groups

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.

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>
```

Example

```
bind lb group Vserver-Group-1 Vserver-LB-1 -persistenceType CookieInsert
```

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>
```

Example

```
set lb group vserver-Group-1 -PersistenceBackup SourceIP -persistMask 255.255.255.255
```

Configuring RADIUS Load Balancing with Persistence

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

Updated: 2013-08-29

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

Updated: 2013-09-13

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>`

- o show lb vserver <name>

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:

- o add cs vserver <name> RADIUS <IP address> <port> -lbmethod TOKEN -rule <rule>
- o show cs vserver <name>

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
```

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

Updated: 2013-09-17

After configuring your virtual servers, you must next configure two services, one for each of the virtual servers that you created. For instructions, see [Configuring Services](#).

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.

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

Updated: 2013-09-17

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. For instructions, see [Configuring Persistence Groups](#).

Viewing Persistence Sessions

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>]
```

Example

```
show lb persistentSessions myVserver
```

To view persistence sessions by using the configuration utility

Navigate to Traffic Management > Virtual server persistent sessions.

Clearing Persistence Sessions

You might need to clear persistence sessions from the NetScaler 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.

Note: The functionality for clearing a particular session that is associated with a given virtual server is available only on NetScaler 10.e.

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>`

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

```
> clear persistentSessions lbvip1
Done
> show persistentSessions
Done
>
```

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
>
```

To clear persistence sessions by using the configuration utility

1. Navigate to Traffic Management > Clear Persistent Sessions.

Overriding Persistence Settings for Overloaded Services

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 `skippersistency` 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>`

Example

```
> set lb vserver mylbvserver -skippersistency ReLb
Done
> show lb vserver mylbvserver
      mylbvserver (*:*) - ANY          Type: ADDRESS
      . . .
      . . .
Skip Persistence: ReLb
      . . .
Done
>
```

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

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 request: 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.

Customizing a Load Balancing Configuration

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](#)

Customizing the Hash Algorithm for Persistence across Virtual Servers

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 NetScaler command line 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 command line interface

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
```

To change the use-port-number global setting by using the configuration utility

1. Navigate to Traffic Management > 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 command line interface

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>
```

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

To specify a hash identifier for an existing service by using the command line interface

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 <serviceName> <memberName> <port> -hashId <positive_integer>
```

```
show servicegroup <serviceName>
```

Example

```
bind servicegroup http_svc_group 10.102.27.153 80 -hashId 2222222

>show servicegroup SRV
SRV - HTTP
State: ENABLED    Monitor Threshold : 0
â€œ|

1)          1.1.1.1:80    State: DOWN    Server Name: 1.1.1.1    Server ID: 123    Weig
Hash Id: 32211

          Monitor Name: tcp-default    State: DOWN
â€œ|

2)          2.2.2.2:80    State: DOWN    Server Name: 2.2.2.2    Server ID: 123    Weig
Hash Id: 12345

          Monitor Name: tcp-default    State: DOWN
â€œ|

Done
```

To specify a hash identifier for a service by using the configuration utility

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 configuration utility

1. Navigate to Traffic Management > Load Balancing > Service Groups.
2. Open a member and type a unique hash ID.

Configuring the Redirection Mode

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 [Networking](#).

To configure the redirection mode by using the command line interface

At the command prompt, type:

```
set lb vserver <name>@ -m <RedirectionMode>
```

Example

```
set lb vserver Vserver-LB-1 -m MAC
```

To configure the redirection mode by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server and select the redirection mode.

Configuring per-VLAN Wildcarded Virtual Servers

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 command line interface

At the command prompt, type the following commands to configure a wildcarded virtual server that listens to a specific VLAN and verify the configuration:

- o add lb vserver <name> <serviceType> IPAddress * Port * -listenpolicy <expression> [-listenpriority <positive_integer>]
- o show vserver

Example

```
add lb vserver Vserver-LB-vlan1 ANY -listenpolicy "CLIENT.VLAN.ID.EQ(2)" -listenpriority 10  
  
show vserver Vserver-LB-vlan1
```

To configure a wildcarded virtual server that listens to a specific VLAN by using the configuration utility

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](#).

Assigning Weights to Services

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 .
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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -weight <Value> <ServiceName>
```

Example

```
set lb vserver Vserver-LB-1 -weight 10 Service-HTTP-1
```

To configure a virtual server to assign weights to services by using the configuration utility

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.

Configuring the MySQL and Microsoft SQL Server Version Setting

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 command line interface

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>`

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
>
```

To set the MySQL Server version parameter by using the command line interface

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>`

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
>
```

To set the MySQL or Microsoft SQL Server version parameter by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. Open a virtual server of type MySQL or MSSQL, and set the server version.

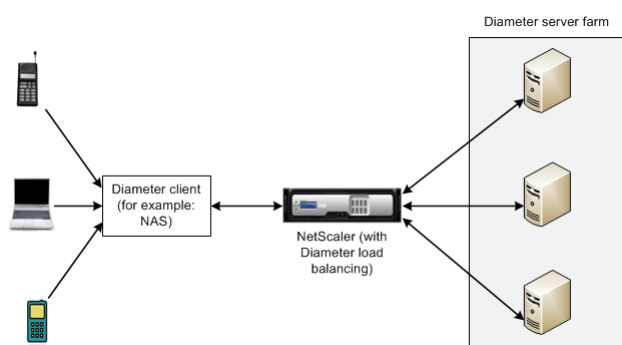
Configuring Diameter Load Balancing

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:



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 connectionâ€”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 Diameter client opens a connection to the NetScaler appliance and sends a Diameter capability exchange (CER) message. Diameter messages are composed of command codes and each command has a set of Attribute-Value Pairs (AVPs), such as Origin-Host and Host-IP-Address.

The NetScaler selects a Diameter server, opens a connection to the server, and forwards the CER message to the server. The server reads the client identity and determines that it is directly connected to the client.

The Diameter server prepares the Diameter handshake reply and sends it to the NetScaler appliance. The appliance modifies the handshake and inserts its own identity. At this point, the Diameter client determines that it is directly connected to the NetScaler (the agent).

Note: Until the Diameter handshake is complete, all Diameter request messages from the client are queued on the selected server. The packets are forwarded to the server when the handshake is complete.

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 NetScaler 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 NetScaler 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.
- NetScaler adds a route entry by putting its identity, because the client sees the NetScaler as a relay agent.

Note: If a Diameter message spans more than one packet, the NetScaler 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 NetScaler splits the packet and forwards the messages to servers as determined by the load balancing virtual server.

Disconnecting 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 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 NetScaler ignores them. The client sends a FIN, which the NetScaler broadcasts to all servers.
- When the NetScaler 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 NetScaler replies with FIN/ACK and removes connections from the reuse pool.
- If the NetScaler 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 NetScaler 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.

Configuring Load Balancing for Diameter Traffic

Updated: 2013-11-12

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

1. Configure the Diameter parameters.

```
set ns diameter -identity <string> -realm <string> -serverClosePropagation <YES|NO>
```

Example

```
set ns diameter -identity mydomain.org -realm org -serverClosePropagation YES
```

2. Add a Diameter monitor.

```
add lb monitor <monitorName> DIAMETER -originHost <string> -originRealm <string>
```

Example

```
add lb monitor diameter_mon DIAMETER -originHost mydomain.org -originRealm org
```

3. Create the Diameter services.

```
add service <name>@ <IP>@ DIAMETER <port>
```

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
```

4. Bind the Diameter services to the Diameter monitor.

```
bind service <name>@ monitorName <monitorName>
```

-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
```

5. Add a Diameter load balancing virtual server with Diameter persistence.

```
add lb vserver <name>@ DIAMETER <IPAddress> <port> -persistenceType DIAMETER -persistAVPno
<positive_integer>
```

Example

```
add lb vserver diameter_vs DIAMETER 10.102.112.152 3868 -persistenceType DIAMETER -pers
```

6. Bind the Diameter services to the Diameter load balancing virtual server.

```
bind lb vserver <name>@ <serviceName>
```

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
```

7. Save the configuration.

```
save ns config
```

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.

Protecting a Load Balancing Configuration against Failure

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.

To protect a load balancing configuration against failure, see the following sections:

- [Redirecting Client Requests to an Alternate URL](#)
- [Configuring a Backup Load Balancing Virtual Server](#)
- [Configuring Spillover](#)
- [Connection Failover](#)
- [Flushing the Surge Queue](#)

Redirecting Client Requests to an Alternate URL

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 command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -redirectURL <URLValue>
```

Example

```
set lb vserver Vserver-LB-1 -redirectURL http://www.newdomain.com/mysite/maintenance
```

To configure a virtual server to redirect the client request to a URL 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 specify a redirect URL.

Configuring a Backup Load Balancing Virtual Server

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.

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 command line interface

At the command prompt, type:

```
set lb vserver <vServerName> -backupVserver <BackupVServerName> [-disablePrimaryOnDown]
```

Example

```
set lb vserver Vserver-LB-1 -backupVserver Vserver-LB-2 -disablePrimaryOnDown
```

To set a backup 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 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.

Configuring Spillover

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 of time after the load on the primary falls below the threshold.

Note: In most cases, spillover is triggered if the value associated with the spillover method exceeds the threshold (for example, number of connections). Keep in mind, however, that 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](#).

Configuring a Predefined Spillover Method

Updated: 2013-09-02

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>
```

Example

```
set lb vserver Vserver-LB-1 -soMethod Connection -soThreshold 1000 -soPersistence enabled -s
```

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.

Configuring Policy Based Spillover

Updated: 2013-09-02

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 on the basis of the virtual server's response time, or on the basis of 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

Updated: 2013-09-13

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:

- o add spillover action <name> -action SPILLOVER
- o show spillover action <name>

Example

```
> add spillover action mySoAction -action SPILLOVER
Done
> show spillover action mySoAction
1)      Name:  mySoAction      Action:  SPILLOVER
Done
>
```

Selecting an Expression for the Spillover Policy

Updated: 2013-09-02

In the policy expression, you can use any virtual-server based expression that returns a Boolean value. For example, you could use one of the following expressions:

`SYS.VSERVER("vserver").RESPTIME.GT(<int>)`, `SYS.VSERVER("vserver").STATE.EQ("œœ<string>œœ")`, and `SYS.VSERVER("vserver").THROUGHPUT.LT(<int>)`.

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 being managed by the virtual server equals 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	<code>SYS.VSERVER("\<vserver-name>").CONNECTIONS</code> , used with the <code>GT(int)</code> arithmetic function.
BANDWIDTH	<code>SYS.VSERVER("\<vserver-name>").THROUGHPUT</code> , used with the <code>GT(int)</code> arithmetic function.
HEALTH	<code>SYS.VSERVER("\<vserver-name>").HEALTH</code> , used with the <code>LT(int)</code> arithmetic function.
DYNAMICCONNECTION	<code>SYS.VSERVER("\<vserver-name>").IS_DYNAMIC_LIMIT_REACHED</code> Note: If you implement policy based spillover by using the <code>IS_DYNAMIC_LIMIT_REACHED</code> function, you must also configure the predefined <code>DYNAMICCONNECTION</code> method for the virtual server, so that statistics required for spillover to work are collected.

Configuring a Spillover Policy

Updated: 2013-09-13

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>`

Example

```
> add spillover policy mySoPolicy -rule SYS.VSERVER("v1").RESPTIME.GT(50) -action mySoAction
Done
> show spillover policy mySoPolicy
1)      Name:  mySoPolicy      Rule:  "SYS.VSERVER("\v1\").RESPTIME.GT(50)" Action:  myS
      Comment:  "Triggers spillover when the vserver\'s response time is greater than 50 m
Done
>
```

Binding a Spillover Policy to a Virtual Server

Updated: 2013-09-13

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>`

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

Updated: 2013-09-02

A backup action specifies what to do in the event that the spillover threshold is reached but one or more backup virtual servers 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 until the number of requests falls below the threshold.

To configure a backup action for spillover by using the command line interface

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

- o set lb vserver <name> -soBackupAction <soBackupAction>
- o show lb vserver <name>

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 configuration utility

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

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 by using the SSF framework. If the L2Conn parameter is set, Layer 2 connection parameters are also synchronized with the secondary.

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.

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.

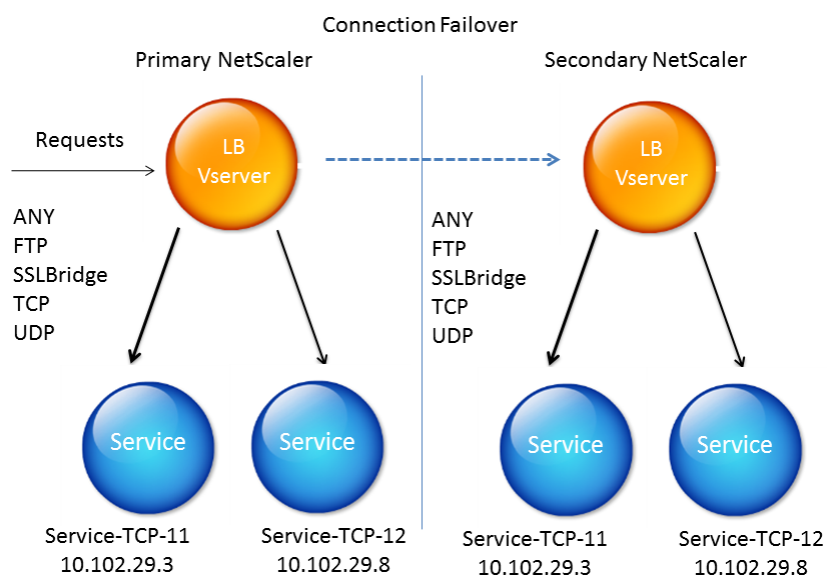
In stateful failover, to maintain current information about the mirrored connections, the primary appliance sends messages to the secondary appliance. The secondary appliance maintains the data related to the packets but uses it only in the event of a failover. If a failover occurs, the new primary (old 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 from the configuration utility.

A basic HA configuration with connection failover contains the entities shown in the following figure.

Figure 1. Connection Failover Entity Diagram



Supported Setup

Connection failover can be configured only on load balancing virtual servers. It cannot be configured on content-switching virtual servers.

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	<p>All methods supported for the service type ANY.</p> <p>However, if Source IP persistence is not set, the SRCIPSRCPORHASH method must be used.</p>	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.	<p>No restriction.</p> <p>It can be ON or OFF.</p>
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
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 NetScaler 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 NetScaler 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.
IPv6 virtual servers	Not yet supported.

To configure connection failover by using the configuration utility, 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 the command line interface, enter the following command:

```
set lb vserver <vServerName> -connFailover <Value>
```

Example

```
set lb vserver Vserver-LB-1 -connFailover stateful
```

When connection failover is disabled on a virtual server, the resources allocated to the virtual server are freed.

To disable connection failover by using the configuration utility, navigate to Traffic Management > Load Balancing > Virtual Servers. Open the virtual server, under Protection, select Connection Failover as Disabled.

To disable connection failover by using the command line interface, enter the following command:

```
set lb vserver <vServerName> -connFailover <Value>
```

Example

```
set lb vserver Vserver-LB-1 -connFailover disable
```

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
Note: 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>]
```

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 SVC
2.

```
flush ns surgeQ
```

The above command flushes all the surge queues on the appliance.

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 Load Balancing Setup

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.

This section includes the following details:

- [Managing Server Objects](#)
- [Managing Services](#)
- [Managing a Load Balancing Virtual Server](#)

Managing Server Objects

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 command line interface

At the command prompt, type:

```
enable server <name>@
```

Example

```
enable server 10.102.29.5
```

To enable or disable a server object by using the configuration utility

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 command line interface

At the command prompt, type:

```
disable server <name>@ <delay>
```

Example

```
disable server 10.102.29.5 30
```

To remove a server object by using the command line interface

At the command prompt, type:

```
rm server <name>@
```

Example

```
rm server 10.102.29.5
```

To remove a server object by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Servers.
2. Select a server, and click Remove.

Managing Services

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>`

Examples

```
enable service Service-HTTP-1
disable service Service-HTTP-1 30
```

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.

Managing a Load Balancing Virtual Server

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 server'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:

- o enable lb vserver <name>@
 - o disable lb vserver
SYNOPSIS
- disable lb vserver <name>

Examples

```
enable lb vserver Vserver-LB-1
disable lb vserver Vserver-LB-1
```

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>
```

Example

```
unbind lb vserver Vserver-LB-1 Service-HTTP-1
```

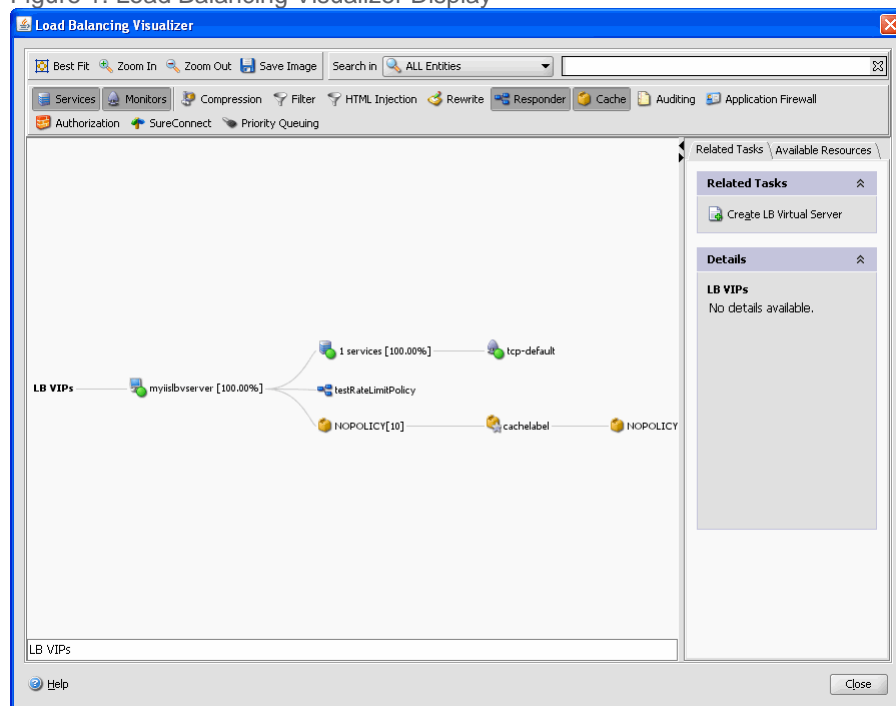
To unbind a service from a virtual server by using the configuration utility

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.

The Load Balancing Visualizer

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



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.
- Load balancing virtual server statistics.
- Statistical information such as the number of requests received per second by the virtual server and the number of hits per second for rewrite, responder, and cache policies.
- A comparative list of all the parameters whose values either differ or are not defined across service containers.

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. Next to the service container is a number that shows the number of services in the group. 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 service container display can help you troubleshoot your configuration if something is not functioning as you expect. More than one container for a particular virtual server is an indication that something is wrong with the configuration of that virtual server and its services. To correct the problem, you must first identify the container that has the desired configuration. You can do so by using the Service Attributes Diff feature, described below. After you identify the container, you right-click the container and click Apply Configuration.

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.
3. In the Load Balancing Visualizer dialog box, you can adjust the viewable area as follows:
 - o Click the Zoom In and Zoom Out icons to increase or decrease the size of the viewed objects. You can click and drag the viewable area if an item that you want to see disappears from view after zooming in.
 - o Click the Best Fit icon to optimize the viewing area.
 - o Click the Save Image icon to save the graph as an image file.
 - o Click the image, hold down the mouse button, and drag the image to pan the view.
 - o In the Search in text field, begin typing the name of the item you are looking for. The item's location is then highlighted. To restrict the search, click the drop-down menu and select the type of element that you want to search for

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, to view configuration details for entities that are bound to this virtual server, you can do the following:
 - o To view a summary of bound services, position the cursor over the virtual server icon.
 - o To view services in a service container, click the icon for a service group, click the Related Tasks tab, click Show Member Services, and then click the service group name. To view additional details about the services click Open.
 - o To view common properties of services in a service group, click the icon for the service group, click the Related Tasks tab, and view the Details section of the tab.
 - o To view a comparative list of the parameters whose values either differ or are not defined across service containers, click the icon for a container, click the Related Tasks tab, and then click Service Attributes Diff. 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.
 - o To view the details for a monitor, position the cursor over the icon or click the icon for the monitor. For additional details, click the icon, click the Related Tasks tab, and then click View Monitor.
 - o To view binding details of a monitor, click the connecting line between the monitor and its related service.

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, to view configuration details for entities that are bound to this virtual server, you can do the following:
 - o To view policies that are bound to this virtual server, select one or more policy icons in the tool bar at the top of the dialog box. For example, you can select Compression, Filter, Rewrite, and Responder. If policy labels are configured, they appear in the main view area.
 - o For bound policies that appear in the view pane of the Visualizer, to view a policy's expression and actions, position the cursor over the policy icon. To view binding details, position the cursor over the line that connects the policy to the virtual server. To view these details, click the policy. The details of the policy appear in the details pane.

To view statistical information 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, to view statistical information, you can do the following:
 - o To view detailed statistics for the load balancing virtual server, click the icon for the virtual server, click the Related Tasks tab, and then click Statistics.
 - o To view the number of requests received per second at a given point in time by the load balancing virtual server 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 and has to be refreshed manually. To refresh this information, click Refresh Stats.

Note: The Show Stats option is available only on NetScaler nCore builds.

To save configuration properties for any entity 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. To copy configuration details for an element to a document or spreadsheet, click the icon for that element, click Related Tasks.
4. In the Related Tasks tab, click Copy Properties and then paste the information into a document.

To bind a resource to 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 for which you want to configure bindings, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, click the Available Resources tab, select a resource type in the drop-down menu, and do one or more of the following:
 - To bind a new monitor to a service, select Monitors, click a particular monitor, and then drag it to the service container icon. Use CONTROL + click to select multiple monitors and drag them to the service.
 - To bind a service or service group, select Services or Service Groups, respectively, click a particular service or service group, and then drag it to the virtual server icon. To bind multiple services or service groups at one time, press CONTROL + click to select multiple services and drag them over the virtual server.
 - To bind a policy, select one of the policy groups, click a particular policy, and then drag it to a virtual server. To bind multiple policies (classic policies only) at one time, press CONTROL + click to select multiple policies and drag them over the virtual server. For details on classic and advanced policies, see [Policy Configuration and Reference](#).

To unbind a resource by using the Visualizer

1. Navigate to Traffic Management > Load Balancing > Virtual Servers.
2. In the details pane, select the virtual server from which you want to unbind a service, policy, or monitor, and then click Visualizer.
3. In the Load Balancing Visualizer dialog box, on the Visualizer image, click the connecting line between the resources that you want to unbind, and then click Unbind. For example, to unbind a monitor, you would click the link between the monitor and its bound service and click Unbind.
4. In the Unbind dialog box, click Yes.

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.

Note: Alternatively, on the Available Resources tab, select the resource type from the drop-down menu, select the particular resource that you want to configure and then click Open.
4. In the modify dialog box, enter new settings for the resource.

To add, remove, or disable 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, right-click the icon for the resource that you want to add, remove, or disable, and then select the corresponding option from the menu.

Note: Alternatively, on the Available Resources tab, click the resource type from the drop-down menu, and then click Add to add an entity, or select the particular resource that you want to configure and then click Open.

Note: These options are not available for service groups or policies.

Managing Client Traffic

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 NetScaler 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, NetScaler always responds.

When you set ICMP VSERVER RESPONSE to ACTIVE on all virtual servers, NetScaler responds even if one virtual server is UP.

When you set ICMP VSERVER RESPONSE to ACTIVE on some and PASSIVE on others, NetScaler responds even if one virtual server set to ACTIVE is UP.

To manage client traffic, see the following sections:

- [Configuring Sessionless Load Balancing Virtual Servers](#)
- [Redirecting HTTP Requests to a Cache](#)
- [Directing Requests According to Priority](#)
- [Directing Requests to a Custom Web Page](#)
- [Enabling Cleanup of Virtual Server Connections](#)
- [Graceful Shut down of Services](#)
- [Rewriting Ports and Protocols for HTTP Redirection](#)
- [Inserting the IP Address and Port of a Virtual Server in the Request Header](#)
- [Using a Specified Source IP for Backend Communication](#)
- [Setting a Timeout Value for Idle Client Connections](#)
- [Managing RTSP Connections](#)
- [Managing Client Traffic on the Basis of Traffic Rate](#)
- [Identifying a connection with Layer 2 Parameters](#)
- [Configuring the Prefer Direct Route Option](#)

Configuring Sessionless Load Balancing Virtual Servers

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 NetScaler 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 NetScaler 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.

Note: 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.

To add a sessionless virtual server by using the command line interface

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>`

- o show lb vserver <name>

Example

```
add lb vserver sesslessv1 any 11.11.12.123 54 -sessionless ENABLED -lbMethod roundrobin -m
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
```

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>
```

Example

```
set lb vserver sesslessv1 -m mac -sessionless ENABLED -lbmethod lrtm
Done
```

To configure a sessionless virtual server by using the configuration utility

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.

Redirecting HTTP Requests to a Cache

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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -cacheable <Value>
```

Example

```
set lb vserver Vserver-LB-1 -cacheable yes
```

To configure cache redirection on 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 Traffic Settings, and select Cacheable.

Directing Requests According to Priority

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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -pq <Value>
```

Example

```
set lb vserver Vserver-LB-1 -pq yes
```

To configure priority queuing on 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 Traffic Settings, and select Priority Queuing.

Note: You must configure priority queuing globally for it to function correctly.

Directing Requests to a Custom Web Page

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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -sc <Value>
```

Example

```
set lb vserver Vserver-LB-1 -sc yes
```

To enable SureConnect on 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 Traffic Settings, and select SureConnect.

Enabling Cleanup of Virtual Server Connections

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 are marked DOWN.

The following table summarizes the effect of this setting on an example configuration consisting of a virtual server, `Vserver-LB-1`, with two services bound to it, `Service-TCP-1` and `Service-TCP-2`. The virtual server intercepts two connections, `C1` and `C2`, and redirects them to `Service-TCP-1` and `Service-TCP-2`, respectively. 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.
E	D	Both client and server connections are terminated. In case of HTTP services, both client and server connections are terminated only if the transaction is active. If the transaction is not active, only client connections are terminated.
D	E	Both client and server connections are terminated. In case of HTTP services, both client and server connections are terminated only if the transaction is active. If the transaction is not active, only server connections are terminated.
D	D	Neither client nor server connections are terminated.

Note: In case of HTTP services, the `downStateFlush` setting is effective only when the client is connected to the server.

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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -downStateFlush <Value>
```

Example

```
set lb vserver Vserver-LB-1 -downStateFlush enabled
```

To configure the down state flush setting on 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 Traffic Settings, and select Down State Flush.

Graceful Shut down of Services

During scheduled network outages such as system upgrades or hardware maintenance, you may have to close or disable some services.

To avoid disrupting established sessions, you can place a service in the TROFS state by doing one of the following:

- Adding a TROFS code or string to the monitorâ€™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 a TROFS-enabled monitor, it can place the service in the TROFS state on the basis of the serverâ€™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 only one monitor must be TROFS-enabled.
- 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 â€"trofsCode <respcode>
add lb monitor <monitor-name> HTTP-ECV â€"trofsString <resp string>
add lb monitor <monitor-name> TCP-ECV â€"trofsString <resp string>
```

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>
```

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 non TROFS-enabled monitor.

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.

Table 1. 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>`

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
```

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.

Rewriting Ports and Protocols for HTTP 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 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.

Redirect URL from the Server	Redirect URL sent to the Client
Only SSL redirect is enabled. The virtual server can be configured on any port.	
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.	
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.	
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.

Redirect URL from the Server	Redirect URL sent to the Client
SSL redirect is enabled on the virtual server. The virtual server can be configured on any port.	
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.	
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.	
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.

Redirect URL from the Server	Redirect URL sent to the Client
The virtual server is configured on port 80.	
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.

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.

Redirect URL from the Server	Redirect URL sent to the Client
------------------------------	---------------------------------

SSL redirect is enabled on the virtual server. The virtual server can be configured on any port.

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.

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 on the virtual server. The virtual server is configured on port 444.

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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -redirectPortRewrite (ENABLED | DISABLED)
```

Example

```
set lb vserver Vserver-LB-1 -redirectPortRewrite enabled
```

To configure HTTP redirection on a virtual server by using the configuration utility

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 command line interface

At the command prompt, type:

- o set ssl vserver <vServerName> - sslRedirect (ENABLED | DISABLED)
- o set ssl service <serviceName> - sslRedirect (ENABLED | DISABLED)

Example

```
set ssl vserver Vserver-SSL-1 -sslRedirect enabled  
set ssl service service-SSL-1 -sslRedirect enabled
```

To configure SSL redirection and SSL port rewrite on an SSL virtual server or service by using the configuration utility

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.

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, 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 command line interface

At the command prompt, type:

```
set lb vserver <name>@ -insertVserverIPPort <insertVserverIPPort> [<vipHeader>]
```

Example

```
set lb vserver Vserver-LB-1 -insertVserverIPPort VipAddr
```

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 > 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.

Using a Specified Source IP for Backend Communication

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. NetScaler 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, NetScaler 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 NetScaler 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 NetScaler can use only the IP addresses accepted by the server as the source IP address.
- The NetScaler 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 NetScaler to use a specified source IP address, create net profiles (network profiles) and configure the NetScaler 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, NetScaler 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, NetScaler uses the profile for all the members of the service group. If a net profile is bound to a monitor, NetScaler 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, NetScaler uses the IP address of the client and ignores all the net profiles. If the USIP option is not enabled, NetScaler selects the source IP in the following manner:

- If there is no net profile on the virtual server or the service/service group, NetScaler uses the default method
- If there is a net profile only on the service/service group, NetScaler uses that net profile.
- If there is a net profile only on the virtual server, NetScaler uses the net profile.
- If there is a net profile both on the virtual server and service/service group, NetScaler uses the net profile bound to the service/service group.

Usage of a net profile for sending monitor probes:

For monitor probes, NetScaler selects the source IP in the following manner:

- If there is a net profile bound to the monitor, NetScaler 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, NetScaler uses the net profile of the service/service group.
 - If there is no net profile even on the service/service group, NetScaler uses the default method of selecting a source IP.

Note: If there is no net profile bound to a service, NetScaler 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. 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 NetScaler 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 NetScaler.

Managing 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 more information on the use of net profiles, see [Using a User-specified Source IP Address for Backend Communication](#).

- 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 Network Profile](#).

Creating an IP Set

Updated: 2014-06-17

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. For more information about the use of IP sets, see [Using a User-specified Source IP Address for Backend Communication](#).

To create an IP set by using the command line interface

At the command prompt, type the following commands:

- `add ipset <name>`
- `bind ipset <name> <IPAddress>@`
or
- `bind ipset <name> <IPAddress>@`
- `show ipset [<name>]`

The above command shows the names of all the IP sets on the NetScaler if you do not pass any name. It shows the IP addresses bound to the specified IP set if you pass a name.

Examples

- ```
> add ipset skpnwipset
Done
> bind ipset skpnwipset 21.21.20.1
Done
```
- ```
> 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
```
- ```
> bind ipset skipipset 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]
> add ns ip 11.11.11.101 255.255.255.0 -type SNIP
ip "11.11.11.101" added
Done
> bind ipset skipipset 11.11.11.101
IPAddress "11.11.11.101" bound
Done
```
- ```
> sh ipset
1) Name: ipset-1
2) Name: ipset-2
3) Name: ipset-3
4) Name: skpnewipset
Done
```
- ```
> 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
```

## To create an IP set by using the configuration utility

Navigate to System > Network > IP Sets, and create an IP set.

## Creating a Net Profile

Updated: 2014-06-17

A net profile (network profile) consists of one or more SNIP or VIP addresses of the NetScaler. For more information about the usage of net profiles, see [Using a User-specified Source IP Address for Backend Communication](#).

### To create a net profile by using the command line interface

At the command prompt, type:

add netprofile <name> [-srcIp <srcIpVal>] 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 baksnip -srcIp bakipset
Done

> set netprofile yahnp -srcIp 12.12.23.1
Done

> set netprofile citkbnip -srcIp citkbipset
Done
```

## Binding a Net Profile to a NetScaler Entity

Updated: 2013-11-12

A net profile can be bound to a load balancing virtual server, service, service group, or a monitor. For more information about the effect of binding a net profile to a NetScaler entity, see [Using a User-specified Source IP Address for Backend Communication](#).

Note: You can bind a net profile at the time of creating a NetScaler entity or bind it to an already 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:

- o set lb vserver <name>@ -netProfile <net\_profile\_name>  
or
- o set cs vserver <name> -netProfile <net\_profile\_name>

#### Examples

```
set lb vserver skpnwvs1 -netProfile gntnp
Done
set cs vserver mmdcsv -netProfile mmdnp
Done
```

### To bind a net profile to 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 Profiles, and set a net profile.

## To bind a net profile to a service by using the command line interface

At the command prompt, type:

```
set service <name>@ -netProfile <net_profile_name>
```

### Example

```
set service brnssvc1 -netProfile brnsnp
Done
```

## To bind a net profile to a service by using the configuration utility

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 command line interface

At the command prompt, type:

```
set servicegroup <serviceName>@ -netProfile <net_profile_name>
```

### Example

```
set servicegroup ndhsvcgrp -netProfile ndhnp
Done
```

## To bind a net profile to a service group by using the configuration utility

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 command line interface

At the command prompt, type:

```
set monitor <monitor_name> -netProfile <net_profile_name>
```

### Example

```
set monitor brnsecvmon1 -netProfile brnsmonnp
Done
```

## To bind a net profile to a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Open a monitor, and set the net profile.

## Setting a Timeout Value for Idle Client Connections

You can configure a virtual server to terminate any idle client connections after a configured timeout 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 lb vserver <name>@ -cltTimeout <Value>
```

#### Example

```
set lb vserver Vserver-LB-1 -cltTimeout 100
```

### To set a time-out value for idle client connections by using the configuration utility

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.

## Managing RTSP Connections

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 command line interface

At the command prompt, type:

```
set lb vserver <name> @â€"RTSPNAT <ValueOfRTSPNAT>
```

#### Example

```
set lb vserver vserver-LB-1 â€"RTSPNAT ON
```

### To configure RTSP NAT by using the configuration utility

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.

## Managing Client Traffic on the Basis of Traffic Rate

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](#).

## Identifying a connection with Layer 2 Parameters

Generally, to identify a connection, the NetScaler 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 command line interface

At the command prompt, type:

```
add lb vserver <name>@ <serviceType> <IPAddress>@ <port> -l2Conn ON
```

### Example

```
add lb vserver LB-VIP1 HTTP 10.1.1.254 80 -l2Conn ON
```

## To configure the L2 connection option by using the configuration utility

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.

## Configuring the Prefer Direct Route Option

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 NetScaler 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 NetScaler 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 command line interface

At the command prompt, type:

set lb parameter -preferDirectRoute (YES | NO)

#### Example

```
set lb parameter -preferDirectRoute YES
```

### To set the prefer direct route option by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Configure Load Balancing parameters.
2. Select Prefer Direct Route.

## Advanced Load Balancing Settings

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’s “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 Serverâ€™s Leve Slow Start

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 â€œnew serviceâ€ 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:

- o Round robin
- o Least connection
- o Least response time
- o Least bandwidth
- o Least packets
- o LRTM (Least Response Time Method)
- o Custom load

For this functionality, you need to set the following parameters:

- o 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.
- o 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:

- o The actual request rate is less than the new service request rate.
- o The service does not receive traffic for three successive increment intervals.
- o The request rate has been incremented 200 times.
- o 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 â€œper second,â€ 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

| Â                                             | new service<br>request rate = 10<br>req/sec<br><br>(Service3added) | new service<br>request rate = 20<br>req/sec | new service<br>request rate = 40<br>req/sec<br><br>(Service4added) | new service<br>request rate = 0<br>req/sec<br><br>(new services<br>exit slow start<br>mode) |
|-----------------------------------------------|--------------------------------------------------------------------|---------------------------------------------|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| Service1                                      | 115                                                                | 110                                         | 80                                                                 | 60                                                                                          |
| Service2                                      | 115                                                                | 110                                         | 80                                                                 | 60                                                                                          |
| Service3                                      | 10                                                                 | 20                                          | 40                                                                 | 60                                                                                          |
| Service4                                      | -                                                                  | -                                           | 40                                                                 | 60                                                                                          |
| Total req/sec (load<br>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 its 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 |

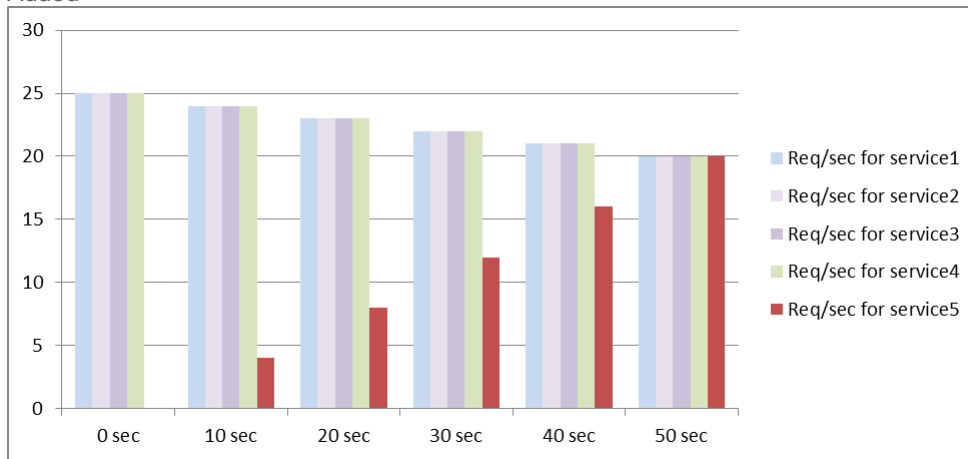
With this configuration, the new service begins receiving as many requests as the existing services 50 seconds after it is added or its state has changed from *DOWN* to *UP*. During each interval in this period, the appliance distributes to the existing servers the excess of requests that would have 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     |
| 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.”

## Setting the Slow Start Parameters

Updated: 2013-12-04

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>`

#### Example

```
> set lb vserver BR_LB -newServiceRequest 5 PER_SECOND -newServiceRequestIncrementInterval 1
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
>
```

### 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

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 NetScaler 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 command line interface

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)
```

### Example

```
>add service nomonsrv 10.102.21.21 http 80
-healthMonitor no
Done
> show service nomonsrv
nomonsrv (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
```

## To set the no-monitor option for an existing service by using the command line interface

At the command prompt, type the following command to set the health monitor option:

```
set service <name> -healthMonitor (YES|NO)
```

### 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
```

## To set the no-monitor option for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open the service, and clear Health Monitoring.

## Protecting Applications on Protected Servers Against Traffic Surges

The NetScaler provides the surge protection option to maintain the capacity of a server or cache. The NetScaler regulates the flow of client requests to servers and controls the number of clients that can simultaneously access the servers. The NetScaler 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 command line interface

At the command prompt, type:

```
set service <name>@ -sp <Value>
```

#### Example

```
set service Service-HTTP-1 -sp ON
```

### To set surge protection on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a source.
2. In Advanced Settings, select Traffic Settings, and select Surge Protection.

## Enabling Cleanup of Service Connections

When cleanup of service connections is enabled, the NetScaler performs a cleanup of the connections on a service that is down. This setting is described in [Enabling Cleanup of Virtual Server Connections](#).

### To set down state flush on the service by using the command line interface

At the command prompt, type:

```
set service <name> -downStateFlush <Value>
```

#### Example

```
set service Service-HTTP-1 -downStateFlush enabled
```

### To set down state flush on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Down State Flush.



## Directing Requests to a Custom Web Page

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 command line interface

At the command prompt, type:

```
set service <name>@ -sc <Value>
```

#### Example

```
set service Service-HTTP-1 -sc ON
```

### To set SureConnect on the service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Sure Connect.

## Enabling Access to Services When Down

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 `accessDown` parameter.

### To enable access down on a service by using the command line interface

At the command prompt, type:

```
set service <name>@ -accessDown <Value>
```

#### Example

```
set service Service-HTTP-1 -accessDown YES
```

### To enable access down on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Access Down.

## Enabling TCP Buffering of Responses

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 command line interface

At the command prompt, type:

```
set service <name>@ -TCPB <Value>
```

#### Example

```
set service Service-HTTP-1 -TCPB YES
```

### To enable TCP Buffering on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select TCP Buffering.

## Enabling Compression

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 command line interface

At the command prompt, type:

```
set service <name> -CMP <YES | NO>
```

#### Example

```
set service Service-HTTP-1 -CMP YES
```

### To enable compression on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Compression.

## Maintaining Client Connection for Multiple Client Requests

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 command line interface

At the command prompt, type:

```
set service <name>@ -CKA <Value>
```

#### Example

```
set service Service-HTTP-1 -CKA YES
```

### To enable client keep-alive on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Client Keep-Alive.

## Inserting the IP Address of the Client in the Request Header

A NetScaler uses the mapped IP address (MIP) 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 NetScaler 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.

### To insert client IP address in the client request by using the command line interface

At the command prompt, type:

```
set service <name>@ -CIP <Value> <cipHeader>
```

#### Example

```
set service Service-HTTP-1 -CIP enabled X-forwarded-for
```

### To insert client IP address in the client request by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Client IP Address.

## Using the Source IP Address of the Client When Connecting to the Server

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."](#)

For information about using the port of the client when connecting to the server, see [Using the Client Port When Connecting to the Server](#).

### To enable USIP mode for a service by using the command line interface

At the command prompt, type:

```
set service <name>@ -usip (YES | NO)
```

#### Example

```
set service Service-HTTP-1 -usip YES
```

### To enable USIP mode for a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Source IP Address.

## Configuring the Source Port for Server-Side Connections

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 NetScaler 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 NetScaler cannot determine which virtual server should receive the response.
  - Use proxy port. When the NetScaler 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.

## Configuring the Use Proxy Port Setting on a Service

Updated: 2013-11-11

You configure the Use ProxyPort setting on the service if you want to override the global setting.

### To configure the Use Proxy Port setting on a service by using the command line interface

At the command prompt, type:

```
set service <name>@ -useProxyPort (YES | NO)
```

#### 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
>
```

### To configure the Use Proxy Port setting on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Traffic Settings, and select Use Proxy Port.

## Configuring the Use Proxy Port Setting Globally

Updated: 2013-09-13



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 command line interface

At the command prompt, type the following commands to configure the Use Proxy Port setting globally and verify the configuration:

- o set ns param -useproxyport ( ENABLED | DISABLED )
- o show ns param

#### Example

```
> set ns param -useproxyport ENABLED
Done
> show ns param
Global configuration settings:
. . .
Use Proxy Port: ENABLED
Done
>
```

### To configure the Use Proxy Port setting globally by using the configuration utility

Navigate to System > Settings > Change global system settings, and select or clear Use Proxy Port.

## Setting a Limit on the Number of Client Connections

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 command line interface

At the command prompt, type:

```
set service <name> -maxclient <Value>
```

#### Example

```
set service Service-HTTP-1 -maxClient 1000
```

### To set a limit to the number of client connections by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Clients.

## Setting a Limit on Number of Requests Per Connection to the Server

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 command line interface

At the command prompt, type:

```
set service <ServiceName> -maxReq <Value>
```

#### Example

```
set service Service-HTTP-1 -maxReq 100
```

### To limit the number of client requests per connection by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Requests.

## Setting a Threshold Value for the Monitors Bound to a Service

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.

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 command line interface

At the command prompt, type:

```
set service <name> -monThreshold <Value>
```

#### Example

```
set service Service-HTTP-1 -monThreshold 100
```

### To set the monitor threshold value on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Monitor Threshold.

## Setting a Timeout Value for Idle Client Connections

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 closes the client connection.

### To set a timeout value for idle client connections by using the command line interface

At the command prompt, type:

```
set service <name> -cltTimeout <Value>
```

#### Example

```
set service Service-HTTP-1 -cltTimeout 100
```

### To set a timeout value for idle client connections by using the configuration utility

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.

## Setting a Timeout Value for Idle Server Connections

You can configure a service with a timeout value to terminate any idle server connections when the configured time 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 command line interface

At the command prompt, type:

```
set service <name>@ -svrTimeout <Value>
```

#### Example

```
set service Service-HTTP-1 -svrTimeout 100
```

### To set a timeout value for idle server connections by using the configuration utility

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.

## Setting a Limit on the Bandwidth Usage by Clients

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 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 command line interface

At the command prompt, type:

```
set service <name> -maxBandwidth <Value>
```

#### Example

```
set service Service-HTTP-1 -maxBandwidth 100
```

### To set a maximum bandwidth limit on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services, and open a service.
2. In Advanced Settings, select Thresholds & Timeouts, and select Maximum Bandwidth.

## Redirecting Client Requests to a Cache

You can configure a service to redirect client requests to a cache, and forward only those requests that are cache misses to a service chosen by the configured load balancing method.

### To set cache redirection on a service by using the command line interface

At the command prompt, type:

```
set service <name>@ -cacheable <Value>
```

#### Example

```
set service Service-HTTP-1 -cacheable YES
```

### To set cache redirection on a service by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Services.
2. Open a service, and set the Cache Type.



## Retaining the VLAN Identifier for VLAN Transparency

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 command line interface

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:

- o set lb vserver <name> -m MAC -macmodeRetainvlan ENABLED
- o show lb vserver <name>

### To configure a load balancing virtual server to retain the client VLAN ID by using the configuration utility

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.

## Configuring Automatic State Transition Based on Percentage Health of Bound Services

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 command line interface

At the command prompt, type the following commands to configure automatic state transition for a virtual server and verify the configuration:

- o `set lb vserver <name>@ -healthThreshold <positive_integer>`
- o `show lb vserver <name>`

### To configure percentage based automatic state transition by using the configuration utility

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 command line interface

At the command prompt, type the following commands to enable the ENTITY-STATE SNMP alarm and verify the configuration:

- o `enable snmp alarm ENTITY-STATE`
- o `show snmp alarm`

### To enable the ENTITY-STATE alarm by using the configuration utility

1. Navigate to System > SNMP > Alarms.
2. Select ENTITY-STATE and, in the Action list, select Enable.

## The Built-in Monitors

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](#).

This section includes the following details:

- [Monitoring TCP-based Applications](#)
- [Monitoring SSL Services](#)
- [Monitoring FTP Services](#)
- [Monitoring SIP Services](#)
- [Monitoring RADIUS Services](#)
- [Monitoring Accounting Information Delivery from a RADIUS Server](#)
- [Monitoring DNS and DNS-TCP Services](#)
- [Monitoring LDAP Services](#)
- [Monitoring MySQL Services](#)
- [Monitoring SNMP Services](#)
- [Monitoring NNTP Services](#)
- [Monitoring POP3 Services](#)
- [Monitoring SMTP Services](#)
- [Monitoring RTSP Servers](#)
- [Monitoring the XML Broker Services](#)
- [Monitoring ARP Requests](#)
- [Monitoring the XenDesktop Delivery Controller Services](#)
- [Monitoring Web Interface Services](#)
- [Monitoring Citrix StoreFront Stores](#)

## Monitoring TCP-based Applications

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                                                                                                                                                                                                                             | <p>The NetScaler appliance establishes a 3-way handshake with the monitor destination, and then closes the connection.</p> <p>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.</p>                                                                                                                                                   |
| http         | <p>httprequest [œ HEAD /œ] - HTTP request that is sent to the service.</p> <p>respcode [200] - A set of HTTP response codes are expected from the service.</p>                                                                             | <p>The NetScaler appliance establishes a 3-way handshake with the monitor destination.</p> <p>After the connection is established, the appliance sends HTTP requests, and then compares the response code with the configured set of response codes.</p>                                                                                                                                                                                                               |
| tcp-ecv      | <p>send ["] - is the data that is sent to the service. The maximum permissible length of the string is 512 K bytes.</p> <p>recv ["] - expected response from the service. The maximum permissible length of the string is 128 K bytes.</p> | <p>The NetScaler appliance establishes a 3-way handshake with the monitor destination.</p> <p>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.</p>                                                                                                                                                                                        |
| http-ecv     | <p>send ["] - HTTP data that is sent to the service</p> <p>recv ["] - the expected HTTP response data from the service</p>                                                                                                                 | <p>The NetScaler appliance establishes a 3-way handshake with the monitor destination.</p> <p>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.</p> |
| ping         | Not Applicable                                                                                                                                                                                                                             | <p>The NetScaler appliance sends an ICMP echo request to the destination of the monitor and expects an ICMP echo response.</p>                                                                                                                                                                                                                                                                                                                                         |

---

To configure built-in monitors for TCP-based applications, see [Configuring Monitors in a Load Balancing Setup](#).

## Monitoring SSL Services

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<br><br>SSL handshake                                              | Successful TCP connection established and successful SSL handshake.                                                                                    |
| HTTP         | TCP connection<br><br>SSL handshake<br><br>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<br><br>SSL handshake<br><br>(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<br><br>SSL handshake<br><br>(Encrypted HTTP request)              | Successful TCP connection is established, successful SSL handshake is performed, and expected HTTP data is received from the server.                   |

## Monitoring FTP Services

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 connections is it marked UP.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

The NetScaler appliance has two built-in monitors for FTP services: the FTP monitor and the FTP-EXTENDED monitor. The FTP monitor checks basic functionality; the FTP-EXTENDED monitor also verifies that the FTP server is able to transmit a file correctly.

| Parameter | Specifies                                                                      |
|-----------|--------------------------------------------------------------------------------|
| userName  | User name used in the probe. Applies to both the FTP and FTP-EXTENDED monitor. |
| password  | Password used in monitoring. Applies to both the FTP and FTP-EXTENDED monitor  |
| fileName  | File name to be used for FTP-EXTENDED monitor only.                            |

To configure built-in monitors to check the state of FTP services, see [Configuring Monitors in a Load Balancing Setup](#).

## Monitoring SIP Services

Note: Support for load balancing SIP traffic over TCP or TLS is available in Netscaler release 10.5.e.

A NetScaler ADC 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: <ul style="list-style-type: none"><li>○ INVITE</li><li>○ OPTION (the default)</li><li>○ REGISTER</li></ul> |
| respcode  | SIP response code with which the SIP service responds the probe request.<br><br>Default: 200.                                                                                                       |



## Monitoring RADIUS Services

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 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](#).

## Monitoring Accounting Information Delivery from a RADIUS Server

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 command line interface

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>`

#### Example

```
add lb monitor radAcctMon RADIUS_ACCOUNTING -radKey "8d#>9jr4rV)L7%a2-zW13sM"
```

## Monitoring DNS and DNS-TCP Services

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 | Parameter                                                                                                                                                                                                                                                                                                                                                                                       |
|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| query     | <p>The DNS query (domain name) sent to the DNS service that is being monitored. Default value: <code>007</code>. If the DNS query succeeds, the service is marked as UP; otherwise, it is marked as DOWN.</p> <p>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.</p> |
| 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](#).

## Monitoring LDAP Services

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 NetScaler IP (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](#).

## Monitoring MySQL Services

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 NetScaler IP (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](#).

## Monitoring SNMP Services

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 NetScaler IP (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](#).

## Monitoring NNTP Services

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 NetScaler IP (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](#).

## Monitoring POP3 Services

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 NetScaler IP (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](#).



## Monitoring SMTP Services

The NetScaler appliance has one 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. If the SMTP service does not respond, or responds incorrectly, it marks the service DOWN.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

| Parameter      | Specifies                                                    |
|----------------|--------------------------------------------------------------|
| userName       | User name SMTP server. This user name is used in the probe.  |
| password       | Password used in monitoring SMTP 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 SMTP monitor, see [Configuring Monitors in a Load Balancing Setup](#).

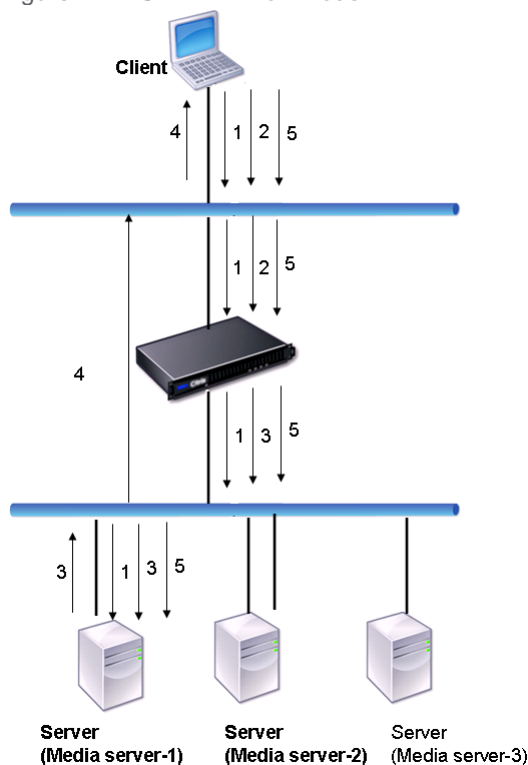
## Monitoring RTSP Servers

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



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.

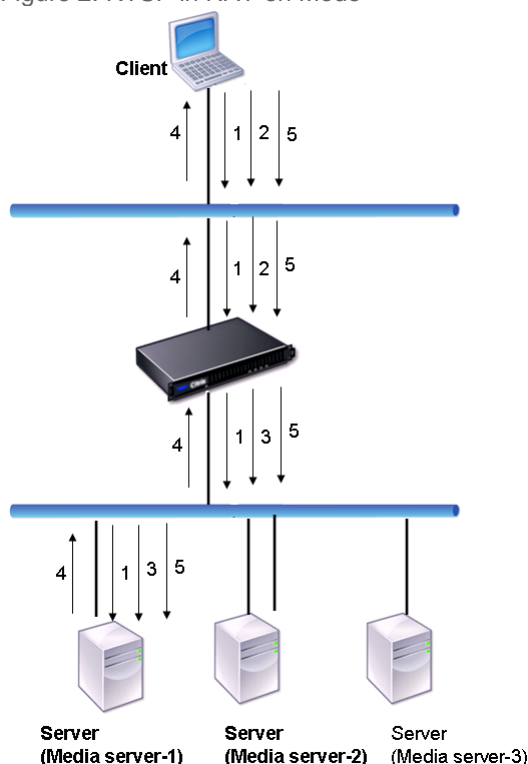
3. Media Server-1 receives the SETUP request from the appliance, allocates resources to process the RTSP request, and sends the appropriate session ID to the client.

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



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:
  - o If the RTSP request comes on the same TCP connection, it routes the request to Media Server-1, maintaining persistence.
  - o 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](#).

## Monitoring the XML Broker Services

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](#)."

## Monitoring ARP Requests

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](#).

## Monitoring the XenDesktop Delivery Controller Services

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 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. If you do not use the wizard, add a monitor of the type CITRIX-XD-DDC.

- For instructions on using the wizard, see [Configuring the load balancing of XenDesktop](#).
- For instructions on adding a monitor, see [Creating Monitors](#).
- For instructions on binding a monitor to a service, see [Binding Monitors to 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>

#### Example

```
> add lb monitor xdddcmon Citrix-xd-ddc -userName Administrator -password E12Dc35450a1 -ddc
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
```

### 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
```

### Example

```
> set lb monitor XD_DDC_21.21.21.22_443_mn CITRIX-xd-ddc -userName Administrator -password
Done
```

## 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.

## Monitoring Web Interface Services

In desktop virtualization, the NetScaler appliance can be used to load balance the Web Interface (WI) servers and Dynamic Desktop Controller (DDC) servers deployed in the Citrix XenApp and Citrix XenDesktop and environments. The NetScaler appliance has two built-in monitor types for monitoring the WI servers used in these environments.

A CITRIX-WEB-INTERFACE monitor can monitor the Web Interface services efficiently because it monitors a dynamic page at the location specified by the site path. The monitor checks for critical failures in resource availability.

To mark a service as UP, the appliance expects the following response from the server:

1. For the first GET request, 200 OK .
2. For the POST request with credentials, 302 Found with the required WIAuthID.
3. For the last GET request with session cookie, 200 OK.

Note: If a redirect URL is configured, 302 Found is expected in the first request before 200 OK.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

When you configure a CITRIX-WEB-INTERFACE monitor, specify the site path to the location of the http page that displays the data collected by the monitor. To monitor the status of the service, in the specified site path, you can view the data updated dynamically by the monitoring script `auth/nocookies.aspx`.

Note: End the site path with a slash (/) to indicate that the monitored resource is dynamic.

Note: When you configure the WI-EXTENDED monitor, when specifying the site path, do not enter a slash (/) at the end of the path as the software internally adds a slash at the end of the path. For example, note the following command:

```
add monitor wi CITRIX-WI-EXTENDED -sitepath "/Citrix/DesktopWeb" -username aaa -password bbb
```

A CITRIX-WI-EXTENDED monitor verifies the logging process with the Web Interface service. This monitor accesses the login page and passes the user name, password, domain, and site path that were specified while configuring the monitor. It verifies the validity of the login credentials, correct configuration of the monitor (for example, the site path), and the connection with the IIS server.

Note: The CITRIX-WI-EXTENDED monitor is supported only for the .NET version of the WI servers. This monitor will not work for the JSP version of the WI servers.

If you use the wizard for configuring load balancing of the XenDesktop servers, a CITRIX-WEB-INTERFACE monitor is automatically created and bound to the WI services. The wizard adds and binds a CITRIX-WEB-INTERFACE monitor by default. If you want to add and bind a CITRIX-WI-EXTENDED monitor, select the Validate Credentials check box and type the necessary data. If you do not use the wizard, add a monitor corresponding to the WI services and bind it to each WI service that you create.

- For instructions on using the wizard, see [Configuring XenDesktop for Load Balancing](#) or [Configuring XenApp for Load Balancing](#).
- For instructions on adding a CITRIX-WEB-INTERFACE monitor, see [Creating Monitors](#).
- For instructions on binding a monitor to a service, see [Binding Monitors to Services](#).

## To add a WI monitor by using the command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> -sitePath <site_path> -dispatcherIP 127.0.0.1 -dispatcherPort 3013 -
userName <username> -password <password> -domain <domain_name>
```

### Examples

```
add lb monitor mwie CITRIX-WEB-INTERFACE -sitePath "/Citrix/XDWI/"
```

```
add lb monitor mwie CITRIX-WI-EXTENDED -sitePath "/Citrix/XDWI/"
-dispatcherIP 127.0.0.1 -dispatcherPort 3013 -userName administrator
-password d83d154575d426 -encrypted -domain wi
```

## To add a WI monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a WI monitor of type CITRIX-WEB-INTERFACE or CITRIX-WI-EXTENDED.



## Monitoring Citrix StoreFront Stores

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, authentication service, and discovery document (in that order). 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.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

Beginning with 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.

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>`

#### Example

```
add lb monitor storefront_ssl STOREFRONT -storename myStore -storefrontacctservice YES -secu
```

### To create a StoreFront monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a WI monitor of type STOREFRONT.

## Custom Monitors

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 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.

This section includes the following details:

- [Configuring HTTP-Inline Monitors](#)
- [Understanding User Monitors](#)
- [How to Use a User Monitor to Check Web Sites](#)
- [Understanding the Internal Dispatcher](#)
- [Configuring a Custom User Monitor](#)
- [Understanding Load Monitors](#)
- [Configuring Load Monitors](#)
- [Unbinding Metrics from a Metrics Table](#)

## Configuring HTTP-Inline Monitors

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:

- o **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.
- o **LOG.** Logs the event in ns/syslog and displays the counters.
- o **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.

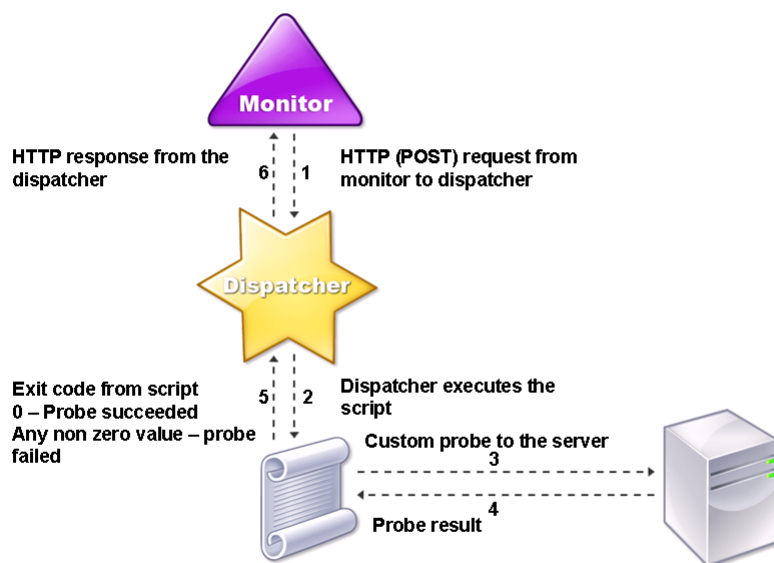
After the service is down, the service remains DOWN for the configured down time. After the DOWN time elapses, the inline monitor uses the configured URL to probe the service to see if it is available again. If the probe succeeds, the state of the service is changed to UP. Traffic is directed to the service, and monitoring resumes as before.

To configure inline monitors, see [Configuring Monitors in a Load Balancing Setup](#).

## Understanding User Monitors

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



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 name of the script file must not exceed 63 characters, and the maximum number of script arguments is 512. To debug the script, you must run it by using the `nsumon-debug.pl` script from the NetScaler command line. 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.

Important: Starting with release 10.5 build 57.x, 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, 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. |
| ^                 | 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. |

|                               |                                                                                                           |
|-------------------------------|-----------------------------------------------------------------------------------------------------------|
| Â                             | Logon fails.                                                                                              |
| POP3                          | Monitor fails to establish a connection to the database.                                                  |
| Â                             | Missing or invalid script arguments, which can include an invalid number of arguments or argument format. |
| Â                             | Logon fails.                                                                                              |
| Â                             | Preparation of SQL query fails.                                                                           |
| Â                             | Execution of SQL query fails.                                                                             |
| SNMP                          | Monitor fails to establish a connection to the database.                                                  |
| Â                             | Missing or invalid script arguments, which can include an invalid number of arguments or argument format. |
| Â                             | Logon fails.                                                                                              |
| Â                             | Monitor fails to create the SNMP session.                                                                 |
| Â                             | Monitor fails to find the object identifier.                                                              |
| Â                             | 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. |
| Â                             | Monitor fails to create a socket.                                                                         |
| Â                             | Mismatch in versions.                                                                                     |
| Â                             | Monitor fails to confirm connection.                                                                      |

You can view the log file from the NetScaler command line 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 NetScaler command prompt:

```
> shell
root@ns# cat /var/nslog/nsumond.log
root@ns# exit
>
```

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.

**Note:** If the user monitor probe times out during high CPU usage, the state of the service remains unchanged.

## How to Use a User Monitor to Check Web Sites

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. |

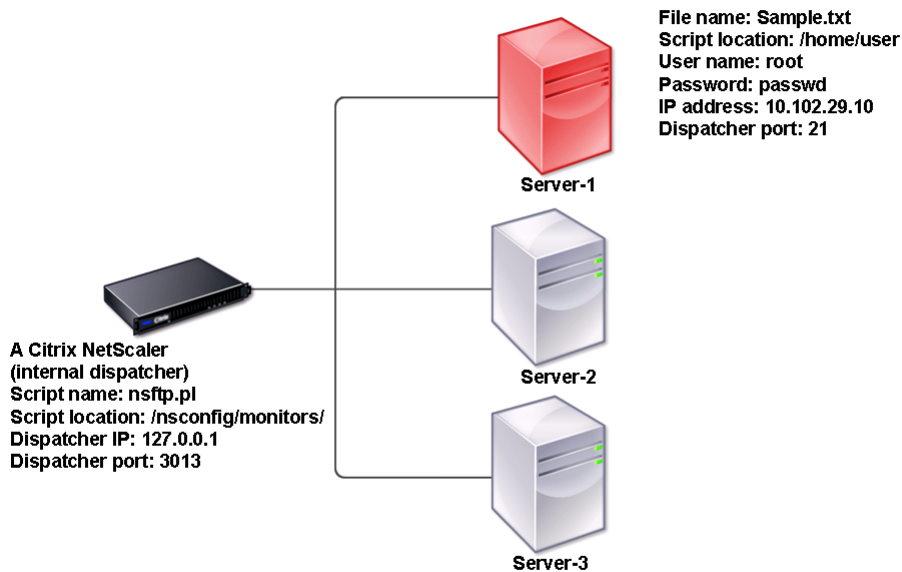
To create a user monitor to monitor HTTP, see [Configuring Monitors in a Load Balancing Setup](#).



## Understanding the Internal Dispatcher

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.

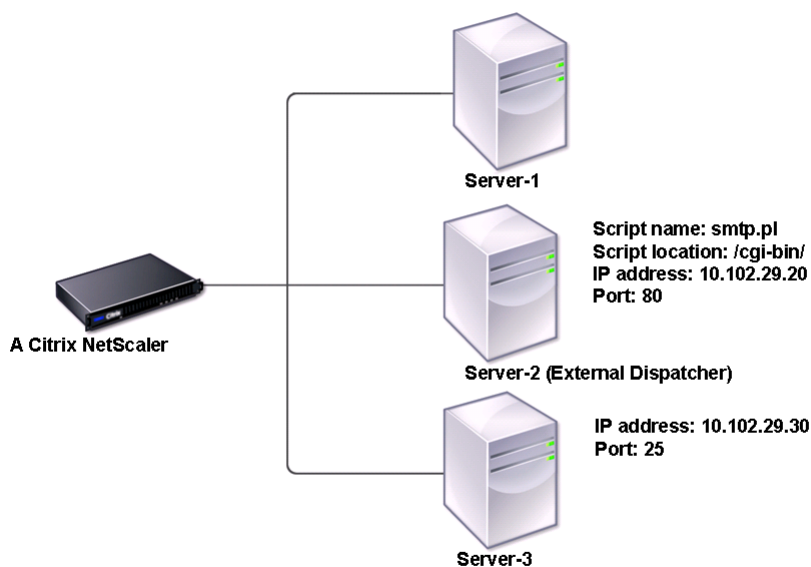
Figure 1. Using a User Monitor with the Internal Dispatcher



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 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



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.

## Configuring a Custom User Monitor

To configure a custom user monitor, you must first write the script that performs the action that the monitor will use to check the service that is bound to it, and upload the script to the /home/user directory on the NetScaler appliance. Then you create the monitor on the appliance, as described below.

Note: Monitor probes originate from the NetScaler IP (NSIP) address.

### 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>
```

#### Example

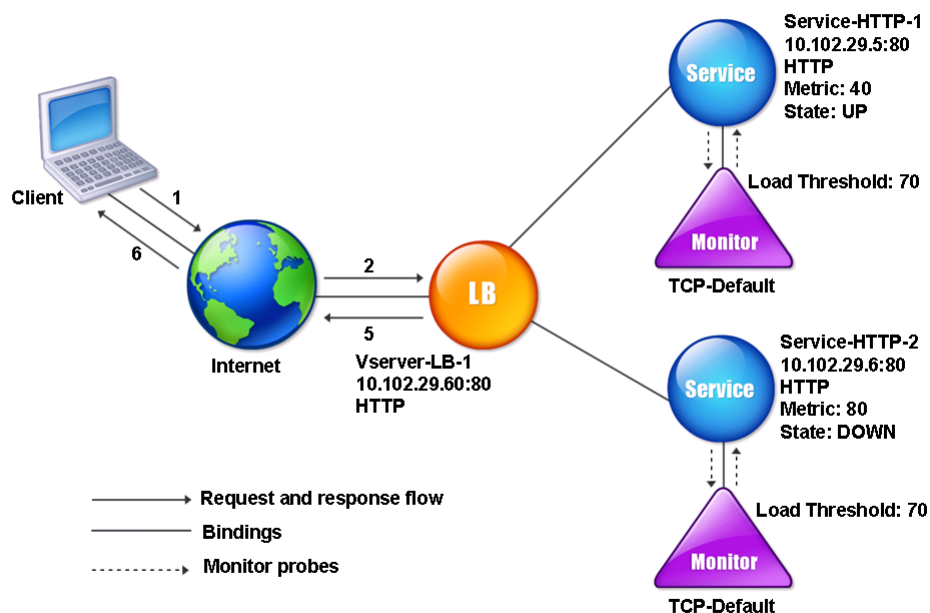
```
add monitor Monitor-User-1 USER -scriptname nsftp.pl "scriptargs "file=/home/user/sample.txt;user=root;password=passwd"
```

## Understanding Load Monitors

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



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.

## Configuring Load Monitors

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>`

#### 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
```

### 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.

## Unbinding Metrics from a Metrics Table

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>
```

#### Example

```
unbind metricTable Table-Custom-1 1.3.6.1.4.1.5951.4.1.1.41.1.5
```

### To unbind metrics from a metric table by using the configuration utility

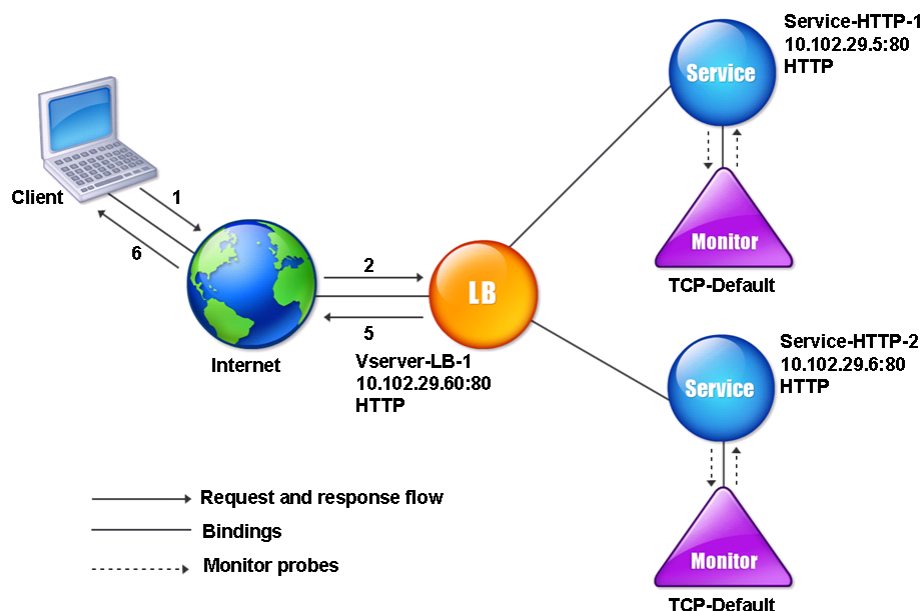
1. Navigate to Traffic Management > Load Balancing > Metric Tables.
2. Open a metric table, select a metric, and click Delete.

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.

## Configuring Monitors in a Load Balancing Setup

To configure monitors on a Web site, 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 following conceptual diagram illustrates a basic load balancing setup with monitors.

Figure 1. How Monitors Operate



As shown above, 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 should fail to respond to the designated number of probes within the designated time period, the monitor marks it DOWN.

This section includes the following details:

- [Creating Monitors](#)
- [Binding Monitors to Services](#)
- [Modifying Monitors](#)
- [Enabling and Disabling Monitors](#)
- [Unbinding Monitors](#)
- [Removing Monitors](#)
- [Viewing Monitors](#)
- [Closing Monitor Connections](#)
- [Ignoring the Upper Limit on Client Connections for Monitor Probes](#)



## Creating Monitors

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 command line interface

At the command prompt, type:

```
add lb monitor <monitorName> <monitorType> [<interval>]
```

#### Example

```
add lb mon monitor-HTTP-1 HTTP
```

```
add lb mon monitor-HTTP-2 TCP 2
```

### To create a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor.

## Binding Monitors to Services

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 all monitors bound to that service using a calculation that you control, and marks the service UP or DOWN depending on the results.

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 lb monitor <monitorName> <ServiceName>
```

#### Example

```
bind mon monitor-HTTP-1 Service-HTTP-1
```

### 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.

## Modifying Monitors

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>
```

#### Example

```
set mon monitor-HTTP-1 HTTP -interval 50 milli
-resptimeout 20 milli
```

### To modify an existing monitor by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and open a monitor to modify.

## Enabling and Disabling Monitors

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>
```

#### Example

```
enable lb mon monitor-HTTP-1
```

### 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>
```

#### Example

```
disable lb mon monitor-HTTP-1
```

## Unbinding Monitors

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>
```

#### Example

```
unbind mon monitor-HTTP-1 Service-HTTP-1
```

### 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.

## Removing Monitors

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>
```

#### Example

```
rm lb monitor monitor-HTTP-1 HTTP
```

### To remove a monitor by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Select a monitor, and click Delete.

## Viewing Monitors

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>
```

#### Example

```
show lb monbindings monitor-HTTP-1
```

### 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>
```

#### Example

```
show lb mon monitor-HTTP-1
```

### 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.

## Closing Monitor Connections

The NetScaler appliance sends probes to the services through the monitors bound to the services. By default, the monitor on the NetScaler 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 monitors, you can configure the NetScaler 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 to all the monitors bound to all the services configured on the NetScaler appliance.

### To configure monitor-connection closure by using the command line interface

At the command prompt, type:

```
set lb parameter -monitorConnectionClose <monitor_conn_close_option>
```

#### Example

```
set lb parameter -monitorConnectionClose RESET
```

### 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.



## Ignoring the Upper Limit on Client Connections for Monitor Probes

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 NetScaler 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)
```

#### Example

```
set lb parameter -monitorSkipMaxClient enabled
```

### 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.

## Managing a Large Scale Deployment

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

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:

- o **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.

- o **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

Updated: 2013-11-12

You create a range of virtual servers as described below.

### To create range of virtual servers by using the command line interface

At the command prompt, type one of the following commands:

- o add lb vserver <name>@ <protocol> -range <rangeValue> <IPAddress> [<port>]
- o add lb vserver <name>@[<rangeValue>]> <protocol> <IPAddress[<rangeValue>]> [<port>]

#### Example

```
add lb vserver Vserver-LB-2 http -range 6 10.102.29.30 80
```

OR

```
> add lb vserver vserver[P-R] http 10.102.29.[26-28] 80
vserver "vserverP" added
vserver "vserverQ" added
vserver "vserverR" added
Done
```

### 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

Updated: 2013-11-12

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>
```

## Example

```
> add service serv[1-3] 10.102.29.[102-104] http 80
service "serv1" added
service "serv2" added
service "serv3" added
Done
```

## Configuring Service Groups

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 NetScaler 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. A domain name will be resolved into an IP address only if the corresponding address record is present on the NetScaler or the name server.

## Creating Service Groups

Updated: 2013-09-04

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>
```

#### Example

```
add servicegroup Service-Group-1 HTTP
```

### To create a service group by using the configuration utility

Navigate to Traffic Management > Load Balancing > Service Groups, and add a service group.

## Binding a Service Group to a Virtual Server

Updated: 2013-11-12

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>
```

#### 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.

## Binding a Member to a Service Group

Updated: 2013-11-12

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 <serviceName> (<IP>@ | <serverName>) <port>
```

### 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 sl.citrite.net
```

## To add members to a service group by using the configuration utility

1. Navigate to **Traffic Management > Load Balancing > Service Groups** and open a service group.
2. Click in the **Service Group** section, and do one of the following:
  - o To add a new IP based service group member, select **IP Based**.
  - o 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**.

## Binding a Monitor to a Service Group

Updated: 2013-12-10

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 <serviceName> -monitorName <string> -monState (ENABLED | DISABLED)
```

### Example

```
bind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
```

## To bind a 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**.

## Managing Service Groups

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.

To manage service groups, see the following sections:

- [Modifying a Service Group](#)
- [Removing a Service Group](#)
- [Unbinding a Member from a Service Group](#)
- [Unbinding a Service Group from a Virtual Server](#)
- [Unbinding Monitors from Service Groups](#)
- [Enabling or Disabling a Service Group](#)
- [Viewing the Properties of a Service Group](#)
- [Viewing Service Group Statistics](#)
- [Load Balancing Virtual Servers Bound to a Service Group](#)

## Modifying a Service Group

Updated: 2013-11-12

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 <serviceName> [-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)]
```

#### 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
```

### 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.

## Removing a Service Group

Updated: 2013-09-04

When you remove a service group, the servers bound to the group retain their individual settings and continue to exist on the NetScaler.

### To remove a service group by using the command line interface

At the command prompt, type:

```
rm servicegroup <ServiceGroupName>
```

#### Example

```
rm servicegroup Service-Group-1
```

### 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.

## Unbinding a Member from a Service Group

Updated: 2013-11-12

When you unbind a member from the service group, the attributes set on the service group will no longer apply to the member that you unbound. The member services retains its individual settings, however, and continues to exist on the NetScaler.

### To unbind members from a service group by using the command line interface

At the command prompt, type:

```
unbind servicegroup <serviceName> <IP>@ [<port>]
```

#### Example

```
unbind servicegroup Service-Group-1 10.102.29.30 80
```

### 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.

## Unbinding a Service Group from a Virtual Server

Updated: 2013-11-12

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>
```

#### Example

```
unbind lb vserver Vserver-LB-1 Service-Group-1
```

### 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.

## Unbinding Monitors from Service Groups

Updated: 2013-12-10

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 <serviceName> -monitorName <string>
```

#### Example

```
unbind serviceGroup Service-Group-1 -monitorName monitor-HTTP-1
```

### 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.

## Enabling or Disabling a Service Group

Updated: 2013-09-04

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>
```

#### Example

```
disable servicegroup Service-Group-1
```

### 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>
```

#### Example

```
enable servicegroup Service-Group-1
```

### 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.

## Viewing the Properties of a Service Group

Updated: 2013-09-04

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, and 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:

- o show servicegroup <ServiceGroupName>
- o show servicegroup <ServiceGroupName> -includemembers

#### Example

```
show servicegroup Service-Group-1
```

### 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

Updated: 2013-09-04

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, such as these, 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>
```

#### Example

```
stat servicegroup Service-Group-1
```

### 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

Updated: 2013-09-04

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:

- o Name
- o State
- o IP address
- o 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>
```

#### 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
>
```

### 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.

## Configuring Automatic Domain Based Service Group Scaling

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:

- `bind serviceGroup <serviceGroupName> <serverName> <port> -autoScale (YES | NO)`
- `show serviceGroup <serviceGroupName>`

### 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.

```
> bind 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:
```

```

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:

 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:

 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:

 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:

 Monitor Name: tcp-default State: UP
 Probes: 2 Failed [Total: 0 Current: 0]
 Last response: Success - TCP syn+ack received.

```

Done

## 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.

## Translating the IP Address of a Domain-Based Server

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>
```

### Example

```
add server myMaskedServer www.example.com -translationIp 10.10.10.10 -translationMask 255.255.0.0 -state ENABLED
```

## 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.

## Masking a Virtual Server IP Address

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>
```

### Example

Pattern matching based on prefix octets:

```
add lb vserver myLBVserver http -ippattern 10.102.0.0 -ipmask 255.255.0.0 80
```

Pattern matching based on trailing octets:

```
add lb vserver myLBVserver1 http -ippattern 0.0.22.74 -ipmask 0.0.255.255 80
```

Modify a pattern-based virtual server:

```
set lb vserver myLBVserver1 -ippattern 0.0.22.74 -ipmask 0.0.255.255
```

## 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.

## Configuring Load Balancing for Commonly Used Protocols

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.

To configure load balancing for commonly used protocols, see the following sections:

- [Load Balancing for a Group of FTP Servers](#)
- [Load Balancing DNS Servers](#)
- [Load Balancing Domain-Name Based Services](#)
- [Load Balancing a Group of SIP Servers](#)
- [Load Balancing RTSP Servers](#)
- [Load Balancing of Remote Desktop Protocol \(RDP\) Servers](#)

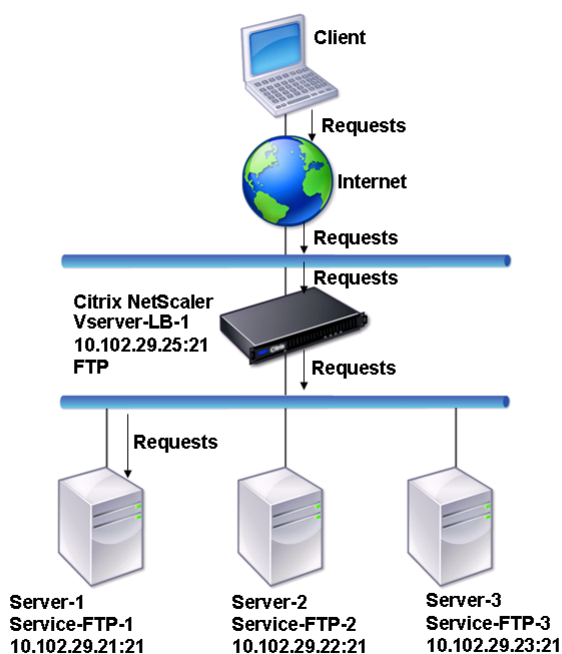


## Load Balancing for a Group of FTP Servers

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



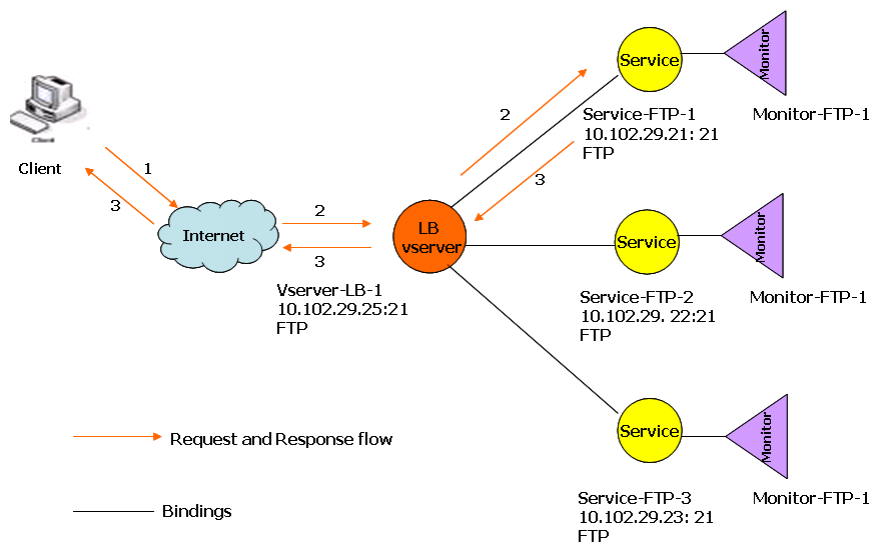
In the diagram, the services Service-FTP-1, Service-FTP-2, and Service-FTP-3 are bound to the virtual server Vserver-LB-1. Vserver-LB-1 forwards the client's connection request to one of the services using the least connection load balancing method. Subsequent requests are forwarded to the service that 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



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>
```

### Example

```
add lb monitor monitor-FTP-1 FTP -interval 360 -userName User -password User
```

## To create FTP monitors by using the configuration utility

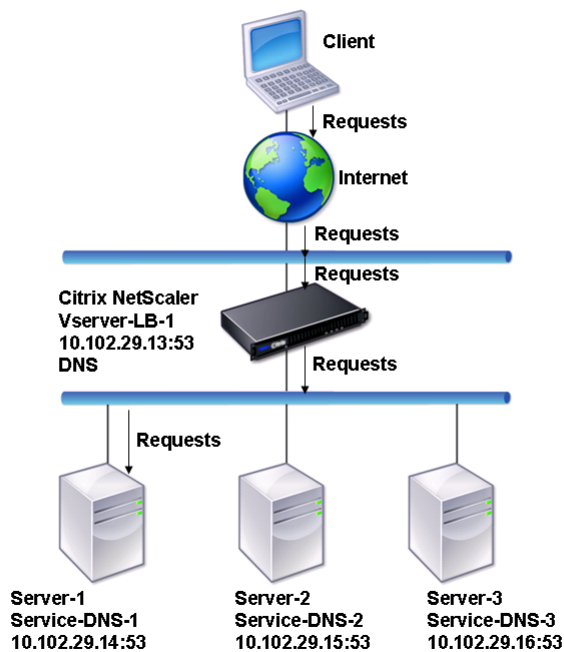
1. Navigate to Traffic Management > Load Balancing > Monitors.
2. Create a monitor of type FTP, and in Special Parameters, specify a user name and password.

## Load Balancing DNS Servers

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

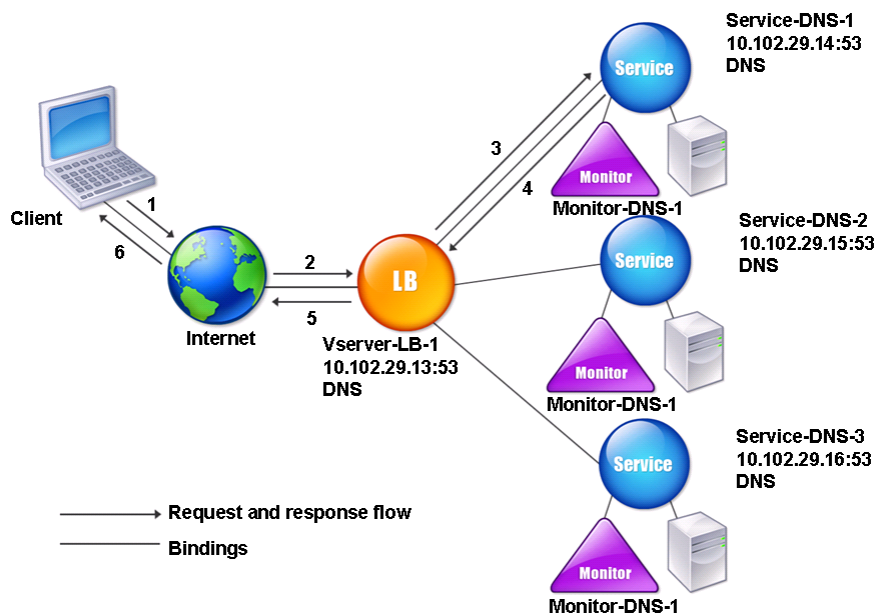


In the diagram, the services Service-DNS-1, Service-DNS-2, and Service-DNS-3 are bound to the virtual server Vserver-LB-1. The virtual server Vserver-LB-1 forwards client requests to a service using the least connection load balancing method. The following table lists the names and values of the basic 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



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>
```

### Example

```
add lb monitor monitor-DNS-1 DNS -query www.citrix.com -queryType Address -IPAddress 10.102.29.14:53
add lb monitor monitor-DNS-2 DNS -query www.citrix2.com -queryType Address -IPAddress 10.102.29.15:53
add lb monitor monitor-DNS-3 DNS -query www.citrix3.com -queryType Address -IPAddress 10.102.29.16:53
```

## 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 Balancing Domain-Name Based Services

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 IPs, if you change the IP address of a server in your load balancing setup, the name server resolves the domain name to the new IP address. The monitor 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.

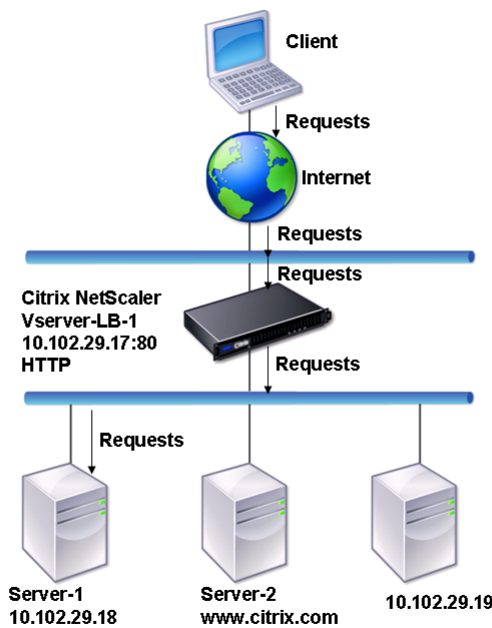
Note: When you change the IP address of a server, the corresponding service is marked down for the first client request. The name server resolves the service IP address to the changed IP address for the next request, and the service is marked UP.

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



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.

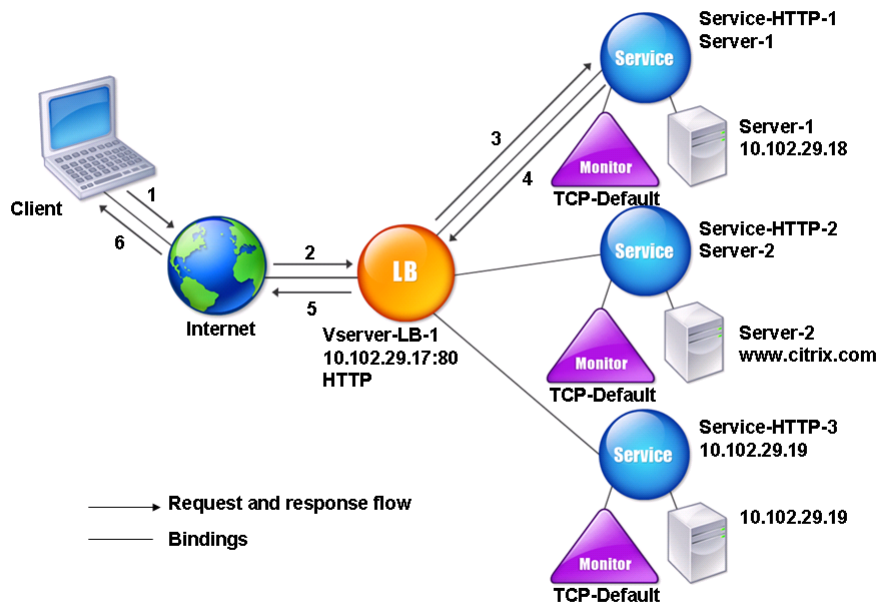
The following table lists the names and values of the basic entities configured on the appliance.

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|--|--|--|--|--|

| Entity type    | Name           | IP address     | Port | Protocol |
|----------------|----------------|----------------|------|----------|
| Virtual Server | Vserver-LB-1   | 10.102.29.17   | 80   | HTTP     |
| Virtual Server | Vserver-LB-2   | 10.102.29.20   | 53   | DNS      |
| Servers        | server-1       | 10.102.29.18   | 80   | HTTP     |
| Servers        | server-2       | www.citrix.com | 80   | HTTP     |
| Services       | Service-HTTP-1 | server-1       | 80   | HTTP     |
| Services       | Service-HTTP-2 | server-2       | 80   | HTTP     |
| Services       | 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



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>
```

### Example

```
add dns nameServer Vserver-LB-2
```

## 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.

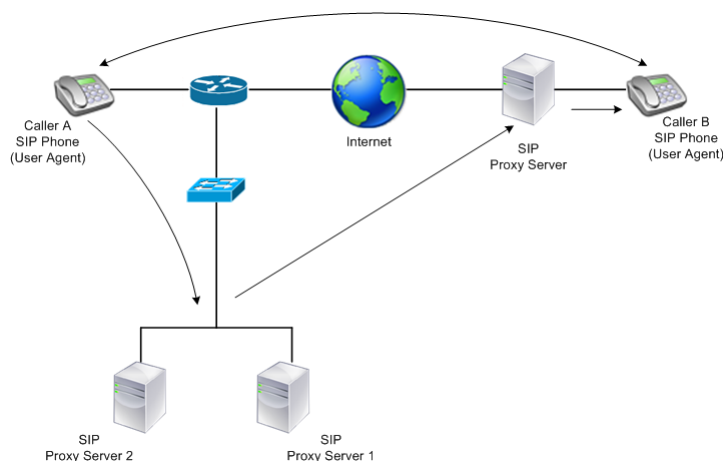
## Load Balancing a Group of SIP Servers

Note: Support for load balancing SIP traffic over TCP or TLS is available in NetScaler release 10.5.e.

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 ADC enables you to load balance SIP messages over UDP or over TCP (including TLS). You can configure the NetScaler ADC 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 ADC 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 ADC 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 ADC 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.

For more information, see [SIP Expressions](#).

## Configuring Load Balancing for SIP Signaling Traffic over TCP or UDP

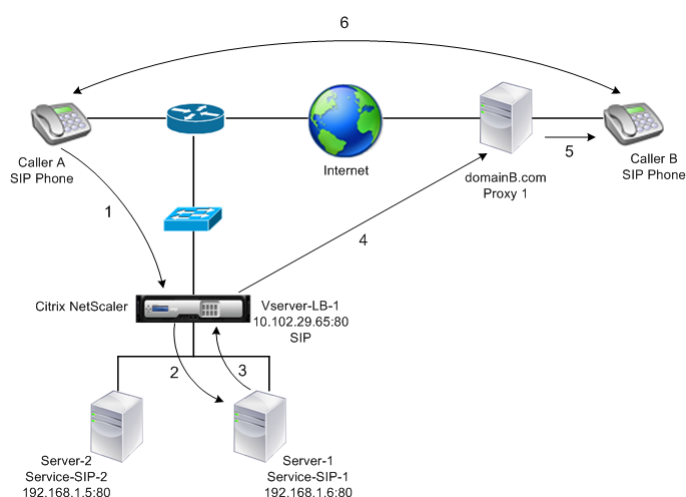


The NetScaler ADC can load balance SIP servers that send requests over UDP or TCP, including TCP traffic secured by TLS. The ADC provides the following service types to load balance the SIP servers:

- o SIP\_UDP â€“ Used when SIP servers send SIP messages over UDP.
- o SIP\_TCP â€“ Used when SIP servers send SIP messages over TCP.
- o SIP\_SSL â€“ Used to secure SIP signaling traffic over TCP by using SSL or TLS. The NetScaler ADC 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



| 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 |
| Ã              | 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:
  - o **SIP\_UDP** â€“ If you are load balancing the SIP traffic over UDP.
  - o **SIP\_TCP** â€“ If you are load balancing the SIP traffic over TCP.
  - o **SIP\_SSL** â€“ 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 ADC 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 ADC 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

1. Create one or more services. At the command prompt, type:

```
add service <name> <serverName> (SIP_UDP | SIP_TCP | SIP_SSL) <port>
```

### Example

```
add service Service-SIP-UDP-1 192.0.2.5 SIP_UDP 80
```

2. 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>
```

### Example

```
add lb vserver Vserver-LB-1 SIP_UDP 10.102.29.60 80
```

3. Bind each service to a virtual server. At the command prompt, type:

```
bind lb vserver <name> <serverName>
```

### Example

```
bind lb vserver Vserver-LB-1 Service-SIP-UDP-1
```

4. (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>
```

### 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
```

5. 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"
```

### Example

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

6. 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 MIP or 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 MIP or 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>
```

### Example

```
set rnat 192.168.1.0 255.255.255.0 -natip 10.102.29.50
```

If you want to use a custom certificate-key pair, bind the custom certificate-key pair to the NetScaler internal service named **nsrnat****sip-127.0.0.1-5061**.

```
add ssl certKey <certkeyName> -cert <string> [-key <string>]
```

```
bind ssl service <serviceName> -certkeyName <string>
```

### Example

```
add ssl certKey c1 -cert cert.epm -key key.ky
```

```
bind ssl service nsrnatsip-127.0.0.1-5061 -certkeyName c1
```

7. 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>
```

## 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

## 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

## 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

## 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 MIP or 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 Request

A NetScaler ADC 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 Request\r\n\r\n"}
```

Done.

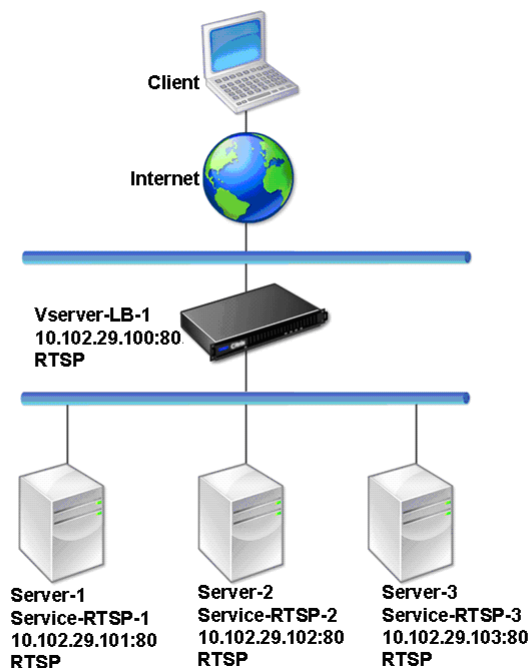
```
> add responder policy sippoll
```

```
> add responder policy sippoll "SIP.REQ.METHOD.EQ(\"NEGOTIATE\")&&SIP.REQ.HEADER(\"Compression\").EXISTS" sipaction1
```

## Load Balancing RTSP Servers

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

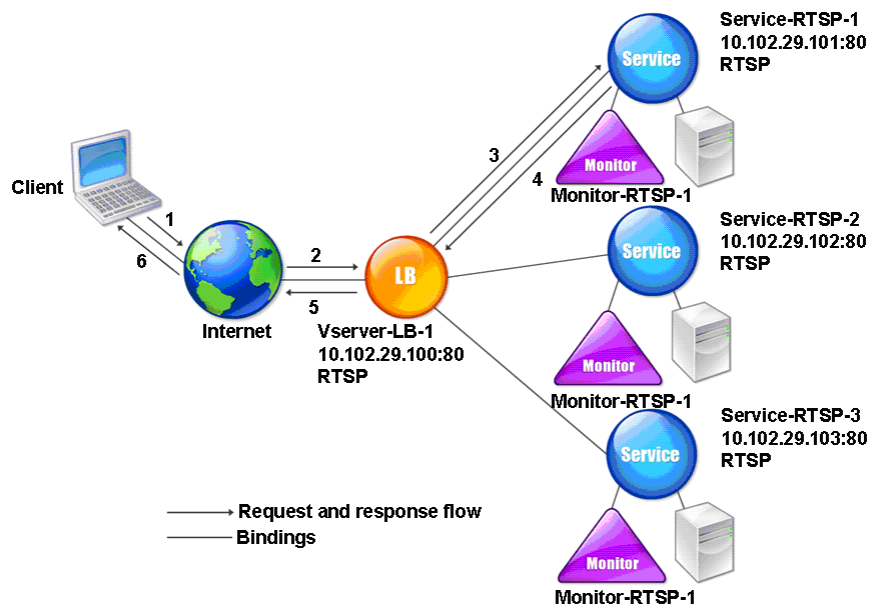


In the example, the services Service-RTSP-1, Service-RTSP-2, and Service-RTSP-3 are bound to the virtual server Vserver-LB-1. The following table lists the names and values of the example entities.

| 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     |
| Â              | Service-RTSP-2 | 10.102.29.102 | 554  | RTSP     |
| Â              | 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



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>
```

### Example

```
add lb monitor Monitor-RTSP-1 RTSP
```

## To configure RTSP monitors by using the configuration utility

Navigate to Traffic Management > Load Balancing > Monitors, and create a monitor of type RTSP.

## Load Balancing of Remote Desktop Protocol (RDP) Servers

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 forwards the response to the client.

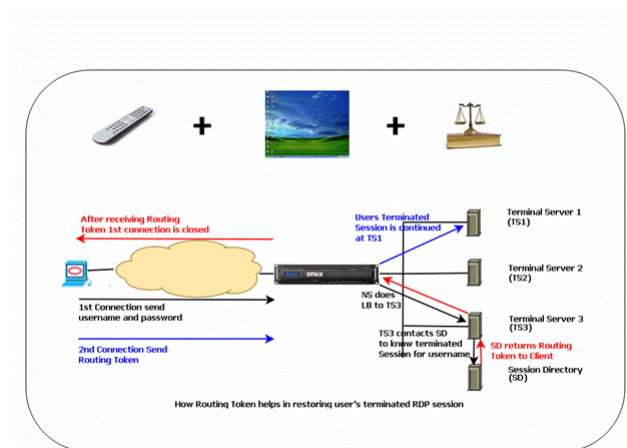
If there is an active session on any other terminal server, the terminal server that receives the request inserts a cookie (referred to as routing token) with the details of the active session and returns the packets to the NetScaler, which returns the packet to the client. The server closes the connection 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



Note: When an RDP service is configured, persistence is automatically maintained by using a routing token. You need not enable persistence explicitly.

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 [http://technet.microsoft.com/en-us/library/cc758177\(WS.10\).aspx#BKMK\\_2](http://technet.microsoft.com/en-us/library/cc758177(WS.10).aspx#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>
```

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
 Monitor Name: tcp-default
 State: UP
 Weight: 1
 Response Time: 4.152 millisec
Done
```

## 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:

- o add lb vserver <name>@ <serviceType> <ipAddress> <port>
- o bind lb vserver <name>@ <serviceName>

Bind all the RDP services to be load balanced to the virtual server.

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
â€¦
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

```

## 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>

### 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

```

## 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

Note: This feature is available in NetScaler release 10.5.e.

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 ADC supports SMPP load balancing and provides optimal distribution of messages across your servers, preventing poor performance and outages.

The NetScaler ADC performs load balancing on the server side when messages are received from clients and on the client side when messages are received from servers.

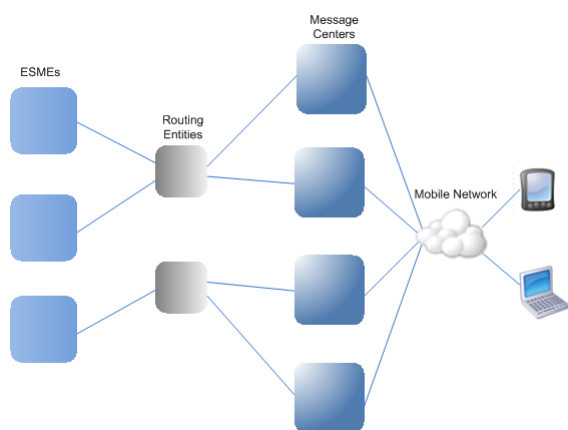
Load balancing of SMPP messages by the NetScaler ADC provides the following benefits:

- 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.

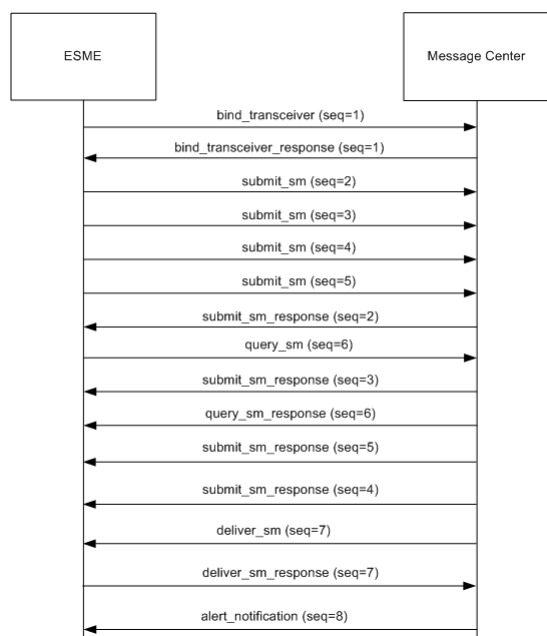
After the connection is established, the ESME can, depending on the mode in which it is bound to the MC, send a submit\_sm or data\_sm message, receive a deliver\_sm or data\_sm message, or send and receive any of these types of messages. The ESME can also send ancillary messages, such as query\_sm, 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.



## How SMPP Load Balancing Works on the NetScaler ADC

Updated: 2014-09-30

An ESME (client) sends a bind message to open a connection to the NetScaler ADC. The ADC authenticates each ESME and, if successful, responds with an appropriate message. The NetScaler ADC 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 ADC 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.

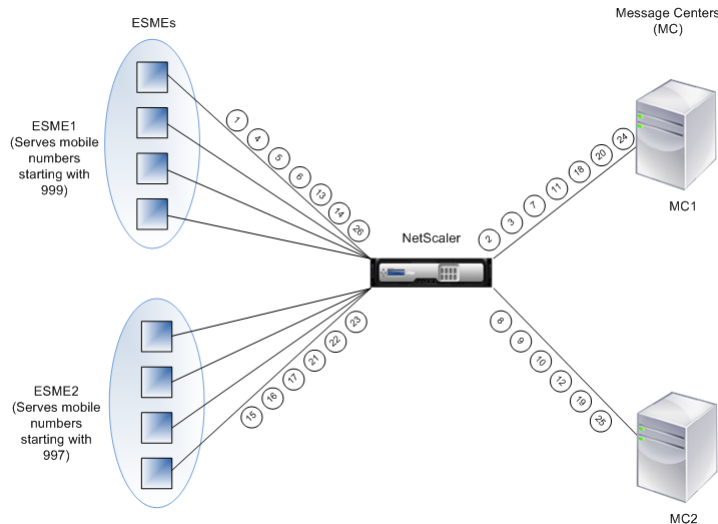
In the original bind request, a client specifies the address range that it can serve. This range is used for forwarding deliver\_sm or data\_sm messages from the message centers to the clients.

### 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 ADC. 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 ADC, and the message centers. For simplicity, only two ESMEs and two message centers are shown.



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 ADC 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:

- o 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 on 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 ADC 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 ADC responds with an error message.

## Configuring SMPP Load Balancing on the NetScaler ADC

Updated: 2014-09-30

Perform the following tasks to configure SMPP load balancing on the ADC:

- Add an SMPP user. The ADC authenticates the user before it accepts a bind request from the user. The user is typically an ESME.
- Add a load balancing virtual server, specifying the protocol as SMPP.
- 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.
- Optionally, create a service group and add services to the service group.
- Optionally, add a monitor of type SMPP-ECV and bind it to the service. A TCP-default monitor is bound by default.
- 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:

- o add smpp user <username> -password <password>
- o add service <name> <IP> SMPP <port> &#x2013;"customserverID <customserverID>
- o add lb vserver <name> <IP> SMPP <port>
- o bind lb vserver <name> <service name>
- o set smpp param

### Example

```
add smpp user smppclient1 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmeth
add smpp user smppclient2 -password c03ebb540695b6110eb31172f32245a1 -encrypted -encryptmeth
add service smmpsvc 10.102.84.140 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED
add service smmpsvc2 10.102.81.175 SMPP 2775 -gslb NONE -maxClient 0 -maxReq 0 -cip DISABLED
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 -addrange "\\d*"
```

### 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: Configuring Rule Based Persistence Based on a Name-Value Pair in a TCP Byte Stream

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(\"49\")"
```

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).TYPECAST_NVLIST_T('=', '^').VALUE("<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  $(n+1)^{\text{th}}$  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: Configuring Load Balancing in Direct Server Return Mode

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:

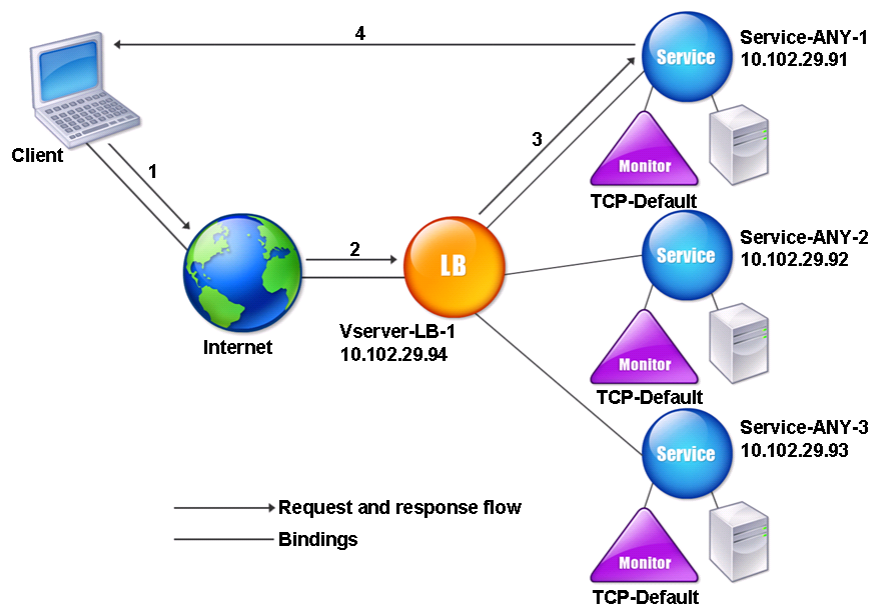
- It supports one-arm mode and inline mode.
- The appliance ages out sessions based on idle timeout.
- Because the appliance does not proxy TCP connections (that is it does not send SYN-ACK to the client), it does not completely shut out SYN attacks. By using the SYN packet rate filter, you can control 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, which it obtains 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 configuration, 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.

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. The following table lists the names and values of the entities configured on the NetScaler 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      |
| Â              | Service-ANY-2 | 10.102.29.92 | ANY      |
| Â              | 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



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. To enable the appliance to do this, you must enable MAC-based forwarding.

#### To enable MAC-based forwarding by using the command line interface

At the command prompt, type:

```
enable ns mode MACbasedforwarding
```

#### To enable MAC-based forwarding by using the configuration utility

1. In the navigation pane, expand System, and then click Settings.
2. On the Settings pane, under Modes and Features, click Configure modes.
3. In the Configure Modes dialog box, select the MAC Based Forwarding check box, and then click OK.
4. In the Enable/Disable Mode(s)? dialog box, click Yes.

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>
```

#### Example

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPHash -m MAC -sessionless enabled
```

#### 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>
```

#### **Example**

```
set service Service-ANY-1 -usip yes
```

### **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: Configuring LINUX Servers in DSR Mode

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
```

Then, run the software that re-maps the TOS id to VIP.

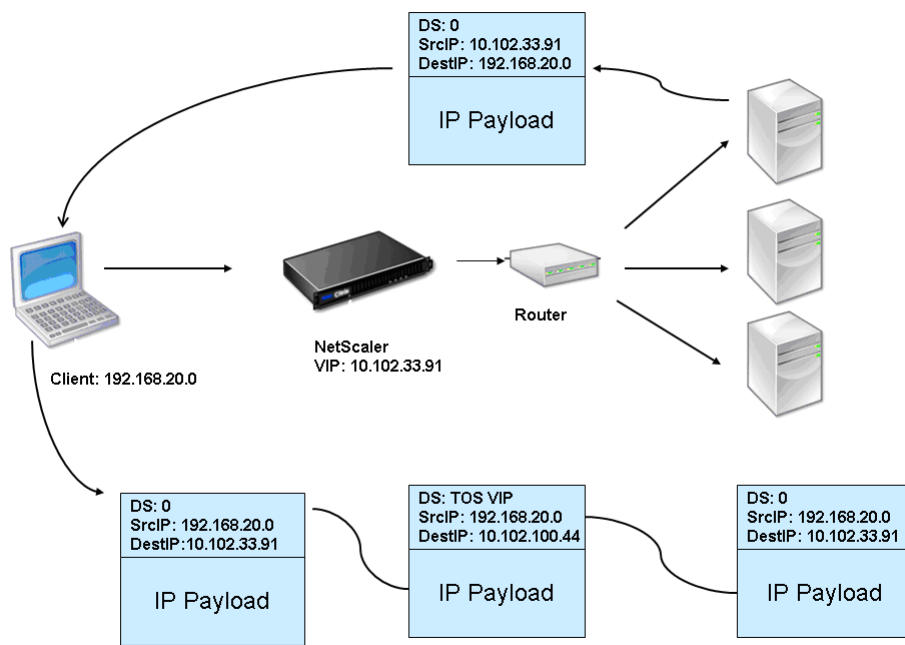
Note: Add the correct mappings to the software before running it. In the preceding commands, the LINUX server uses dummy0 to connect to the network. When you use this command, type the name of the interface that your LINUX server uses to connect to the network.

## Use Case 5: Configuring DSR Mode When Using TOS

Differentiated services (DS), also known as TOS (Type of Service), is a field that is part of the TCP packet header. 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



The TOS feature is specifically 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>
```

### Example

```
set lb vserver Vserver-LB-1 -m TOS -tosId 3
```

## 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>
```

### Example

```
add monitor mon1 PING -destip 10.102.33.91 -tos Yes -tosId 3
```

## 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.

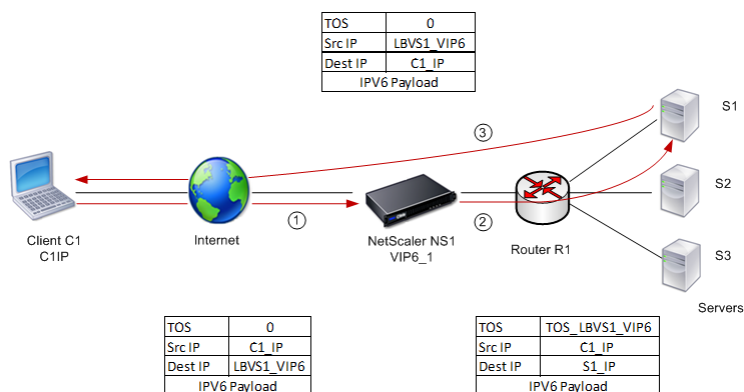
## Use Case 6: Configuring Load Balancing in DSR Mode for IPv6 Networks by Using the TOS Field

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 NetScaler.

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.



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:

- Client C1 sends a request to virtual server LBVS1.
- 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.
- 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 NetScaler, 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:

- o enable ns mode USIP
- o add service <serviceName> <IP> <serviceType> <port>

Repeat the above command as many times as necessary to add each server as a service on the NetScaler appliance.

- o add lb vserver <name> <serviceType> <ip> <port> -m <redirectionMode> -tosld <positive\_integer>
- o bind lb vserver <vserverName> <serviceName>

## 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: Configuring Load Balancing in DSR Mode by Using IP Over IP

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.

### Configuring a Load Balancing Virtual Server

Updated: 2013-11-29

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>`

#### 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
```

#### 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.

### Configuring Services for IP over IP DSR

Updated: 2013-11-29

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>`

- o add monitor <monitorName> <monitorType> -destip <ip> -iptunnel <iptunnel>
- o bind service <serviceName> -monitorName <monitorName>

### 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
```

### 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>
```

### Example

```
bind lb vserver Vserver-LB-1 Service-DSR-1
```

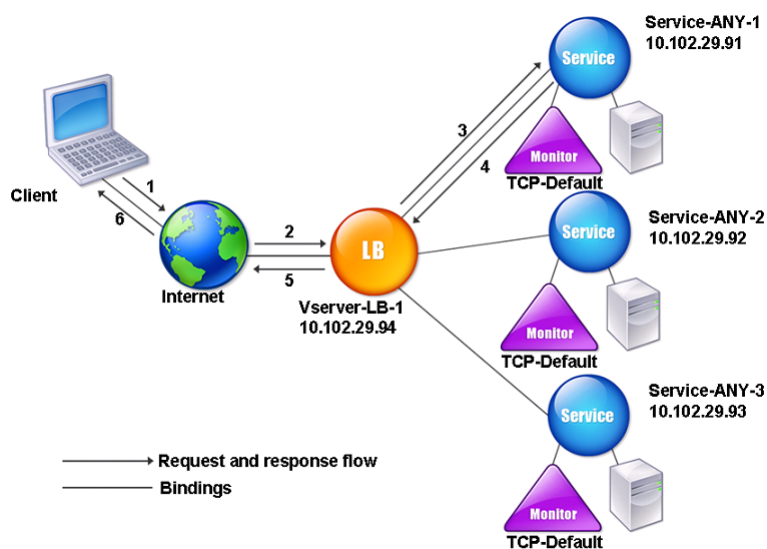
### To bind a service to a load balancing virtual server by using the configuration utility

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.

## Use Case 8: Configuring Load Balancing in One-arm Mode

In a one-arm setup, you connect the NetScaler appliance to the network through a single interface. This is one of the simplest deployment scenarios, where the router, the servers and the appliance are all connected to the same switch. The client can access the server directly, bypassing the appliance, if the client knows the IP address of the server. 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. Entity Model for 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.

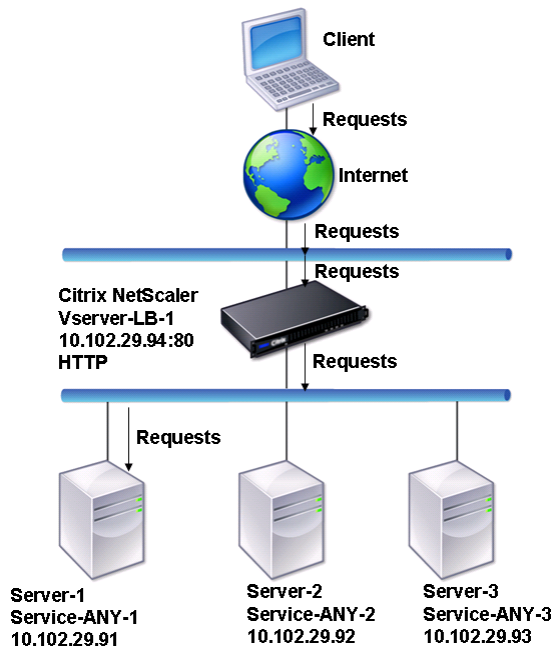
| Entity type    | Name          | IP address   | Protocol |
|----------------|---------------|--------------|----------|
| Virtual server | Vserver-LB-1  | 10.102.29.94 | ANY      |
| Services       | Service-ANY-1 | 10.102.29.91 | ANY      |
| Â              | Service-ANY-2 | 10.102.29.92 | ANY      |
| Â              | 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: Configuring Load Balancing in the Inline Mode

In an inline mode (also called two-arm mode) setup, you deploy the NetScaler appliance to the network through more than one interface. In the two-arm setup, the appliance is connected between the servers and the client. Traffic from clients passes through the appliance to access the load balanced server. 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

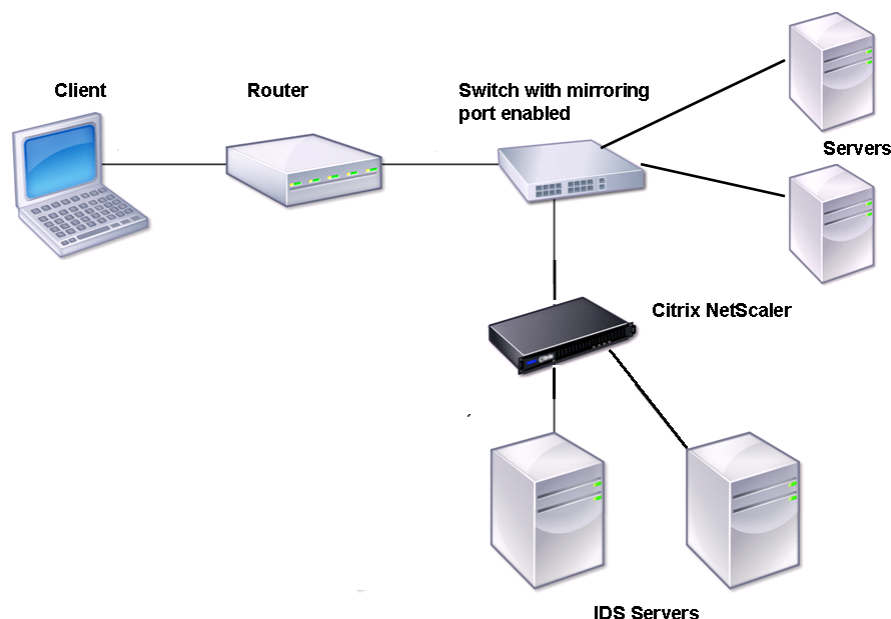


The configuration and the entity diagram for inline mode are the same as described in "[Configuring Load Balancing in One-arm Mode](#)."

## Use Case 10: Load Balancing of Intrusion Detection System Servers

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.

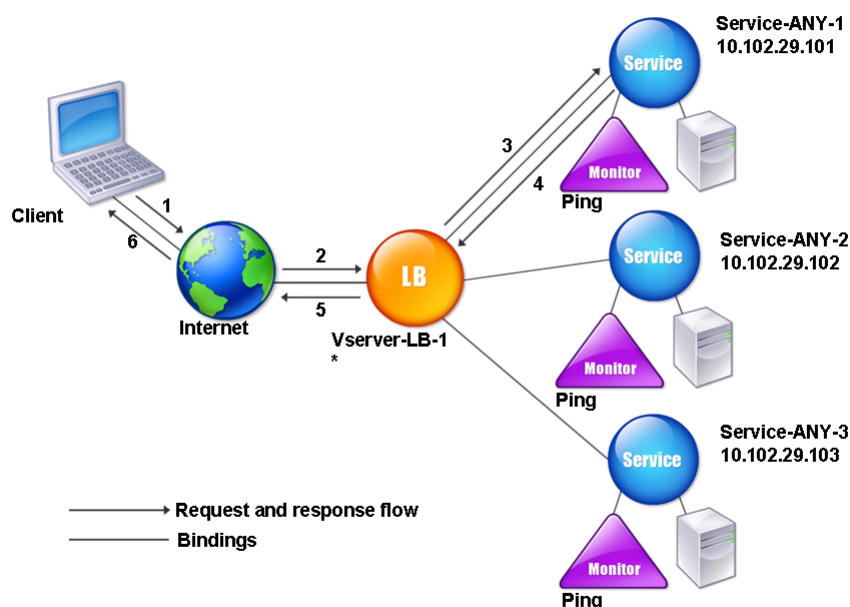
| Entity type    | Name          | IP address    | Port | Protocol |
|----------------|---------------|---------------|------|----------|
| Virtual server | Vserver-LB-1  | *             | *    | ANY      |
| Services       | Service-ANY-1 | 10.102.29.101 | *    | ANY      |
| Â              | Service-ANY-2 | 10.102.29.102 | *    | ANY      |
| Â              |               |               |      |          |

|          |               |               |      |      |
|----------|---------------|---------------|------|------|
|          | 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



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>
```

#### Example

```
enable ns mode MAC
```

#### 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>
```

#### Example

```
set lb vserver Vserver-LB-1 -lbMethod SourceIPDestIPHash -m MAC -sessionless enabled
```

#### **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>
```

#### **Example**

```
set service Service-ANY-1 -usip yes
```

#### **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

A very 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 multi-tenant 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 a number of 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.

Table 1. Client Parameters Used to Define Listen Policies

| 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                                                                              |

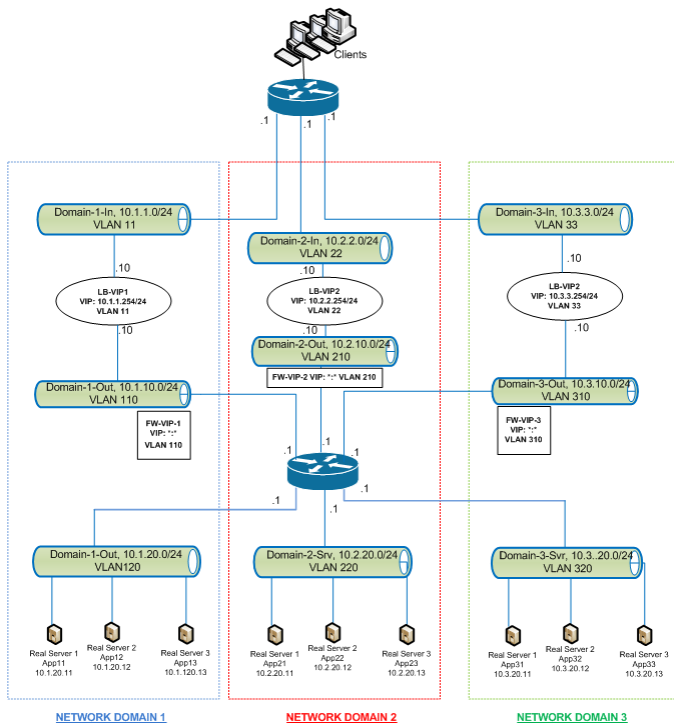
On the NetScaler appliance, a virtual server is configured for each domain, with a listen policy specifying that the virtual server is to listen only to traffic for that domain. Also configured for each domain is a shadow load balancing virtual server, which listens to traffic destined for any domain. 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



## 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:

- o 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 more details, see "[Client Parameters Used to Define Listen Policies](#)."
- o 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 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.

- o Add services representing the server pools in the domain, and bind them to the virtual server.
- o Configure the firewall for each domain as a service, and bind all of 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:

- o add policy expression <expressionName> <listenPolicyExpression>
- o add lb vserver <name> <serviceType> <ip> <port> -l2conn ON -listenPolicy <expressionName>

Add a load balancing virtual server for each domain. This virtual server is for traffic of the same domain.

- o add lb vserver <name> ANY \* \* -l2conn ON -m MAC -listenPolicy <expressionName>

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 e1
add lb vserver FW-VIP-2 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e2
add lb vserver FW-VIP-3 ANY * * -persistenceType NONE -lbMethod ROUNDROBIN - listenPolicy e3

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
```

## **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 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: Configuring XenDesktop for Load Balancing

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.

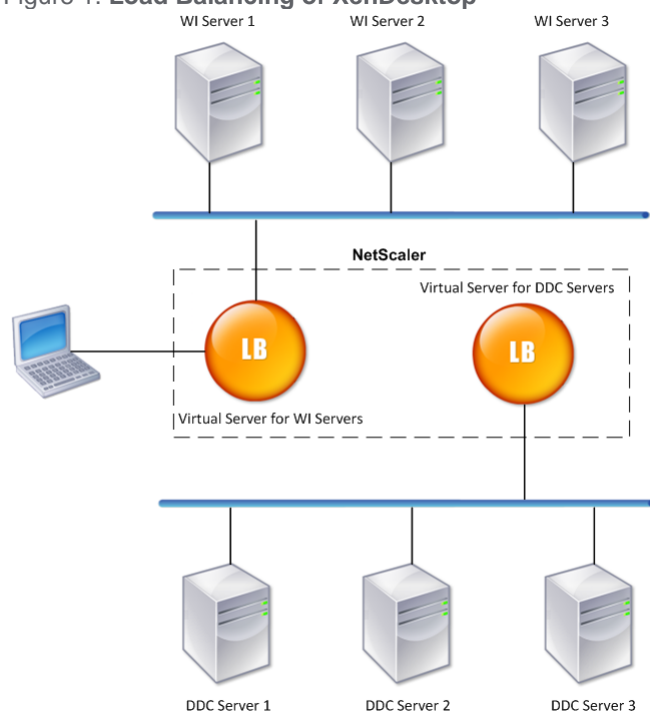
The DDC also authenticates users and manages the assembly of the users' virtual desktop environments by controlling the state of the desktops, and starting and stopping the desktops.

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**



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 DDC servers even though you use the SSL protocol for communication with the client.

A wizard is available for configuring basic load balancing in a XenDesktop deployment. You can use the wizard to configure Web interface servers and a virtual server for them, and DDC servers and a virtual server for them. The virtual servers that you configure are bound to services specified as Web Interface services and DDC services. Each virtual server is configured with the default load balancing method, and the default features are enabled. A monitor is created and bound to each virtual server.

The wizard creates a basic setup, with default values for options such as the load balancing method, policies, persistence, and advanced settings. You can change any of the values if necessary.

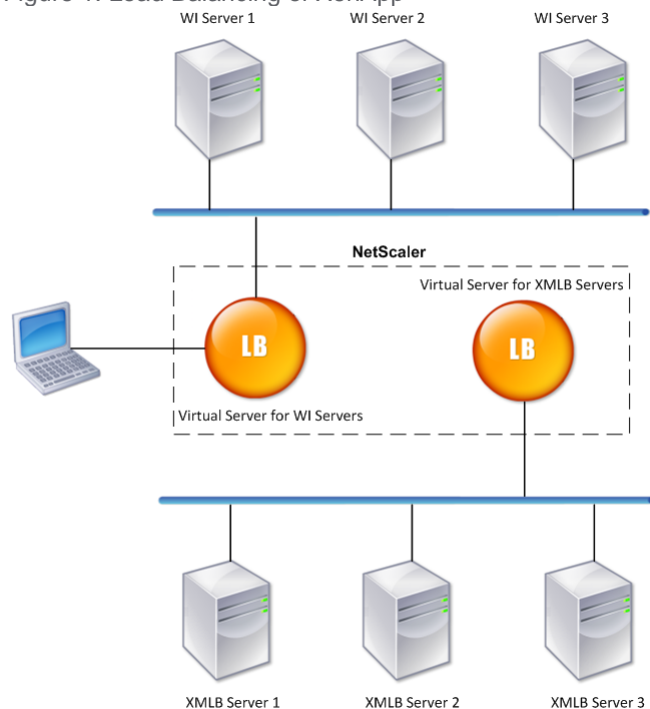
### To configure load balancing for XenDesktop by using the configuration utility

Click XenApp and XenDesktop, and follow the instructions in the wizard.

## Use Case 13: Configuring XenApp for Load Balancing

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



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 NetScaler. 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.

The configuration utility provides a wizard for setting up basic load balancing for XenApp.

Through this wizard, you can configure Web Interface servers and a virtual server for them, and XML Broker servers and a virtual server for them. You can also specify the site through which the status of Web Interface servers can be monitored and the software application used to monitor the status of the XML Broker servers.

When you complete the wizard, a basic load balancing setup is configured on the NetScaler. The specified virtual servers are created and bound to the services specified as Web Interface services and XML Broker services. Each virtual server is configured with the default load balancing method, and the default features are enabled. A monitor is created and bound to each virtual server.

The wizard creates a basic setup with default values for options such as the load balancing method, policies, persistence, and advanced settings. You can change any of the values if necessary.

## To configure load balancing for XenApp by using the configuration utility

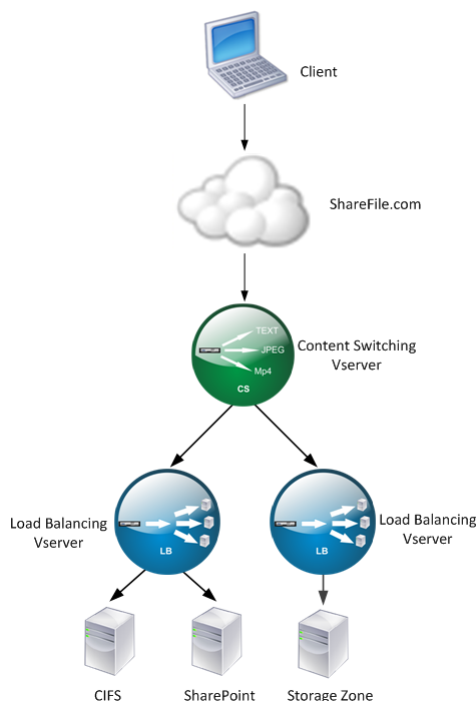
Click XenApp and XenDesktop, and follow the instructions in the wizard.

## Use Case 14: ShareFile Wizard for Load Balancing Citrix ShareFile

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



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

Updated: 2015-06-04

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:
  - o Name: Name of the content switching virtual server.
  - o IP Address: IP address of the content switching virtual server.
  - o 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.



## Setup Load Balancing for ShareFile

ShareFile Configuration

Name\*

IP Address\*

☒ 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.

## Setup Load Balancing for ShareFile

ShareFile Configuration

| Name                        | IP Address   | Port | Protocol | Selected                                       |
|-----------------------------|--------------|------|----------|------------------------------------------------|
| ShareFile CS Virtual Server | 10.102.29.96 | 443  | SSL      | Sharefile Data, Network File Shares/SharePoint |

Certificate

☐ Choose Certificate ☒ Install Certificate

Certificate\*

Key\*

6. Click Continue.
7. In the Add New StorageZone Controller dialog box, specify the values of the following parameters:
- StorageZone Controller IP Addressâ€™ IP address
  - Portâ€™ Port number. The default value is 443.
  - Protocolâ€™ Select from HTTPS or HTTP

ShareFile StorageZone Controller Configuration

Add New StorageZone Controller X

StorageZone Controller IP Address\*  +

Port\*

Protocol\*

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 Addressâ€™ IP address of AAA virtual server
  - LDAP Server IP Addressâ€™ IP Address of the LDAP server
  - Portâ€™ Port number. The default value is 389
  - Time outâ€™ The time out value in minutes
  - Single Sign-on Domainâ€™ Single sign-on domain name
  - Base DNâ€™ Base domain name
  - Administrator Bind DNâ€™ LDAP account name with the domain name, for example, administrator@domainname.com
  - Logon Nameâ€™ Logon name is the sAMAccount name
  - Password and Confirm Passwordâ€™ Enter the password and confirm the password

LDAP Authentication Settings

☒ Configure New

AAAServer IP Address\*

LDAP Server IP Address\*

Port\*

Time out\*

Single Sign-on Domain\*

Base DN (location of users)\*

Administrator Bind DN\*

Logon Name\*

Password\*

Confirm Password\*

.

.

.

.

.

.

.

.

389

3

Cn=Users,dc=example,dc=com

administrator@example.com

sAMAccountName

Continue

Cancel

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

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

Updated: 2013-08-01

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

Updated: 2015-06-11

#### ◦ 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 NetScaler command line 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 donâ€™t need to add any prefix.

- o **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`.

- o **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.

- o **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.

- o **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.

- o **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.

#### o 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 if any service is not stable. If yes, complete the troubleshooting steps mentioned for service not stable issue.

#### o Issue

The service status is displayed as DOWN.

#### Resolution

Consider troubleshooting the following components:

Verify whether a SNIP or MIP 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.

#### o 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.

#### o 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.

#### o 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.

#### o 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.

#### o 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.

## SSL Offload and Acceleration

A Citrix® 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.

## Configuring SSL Offloading

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.

**To configure SSL offloading, see the following sections:**



## Enabling SSL Processing

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.

### To enable SSL processing by using the command line interface

At the command prompt, type:

- enable ns feature ssl
- show ns feature

#### 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)  | <b>SSL Offloading</b> | <b>SSL</b> | <b>ON</b> |
| .   |                       |            |           |
| .   |                       |            |           |
| .   |                       |            |           |
| 24) | NetScaler Push        | push       | OFF       |

Done

### To enable SSL processing by using the configuration utility

Navigate to System > Settings and, in the Modes and Features group, select Configure Basic Features, and select SSL Offloading.

## Configuring 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.

### To add a service by using the command line interface

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>

#### Example

```
> add service ssl1 10.102.29.252 HTTP 80
Done
> show service ssl1
 ssl1 (10.102.29.252:80) - HTTP
 State: UP
 Last state change was at Thu Nov 12 05:26:31 2009
 Time since last state change: 0 days, 00:00:06.750
 Server Name: 10.102.29.252
 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): YES
 Idle timeout: Client: 180 sec Server: 360 sec
 Client IP: DISABLED
 Cacheable: NO
 SC: OFF
 SP: ON
 Down state flush: ENABLED

1) Monitor Name: tcp-default
 State: UP Weight: 1
 Probes: 2 Failed [Total: 0 Current: 0]
 Last response: Success - TCP syn+ack received.
 Response Time: N/A

Done
```

### To modify or remove a service by using the command line interface

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.

### To configure a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, create a service, and specify the protocol as SSL.

## Configuring an SSL-Based Virtual Server

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.

### To add an SSL-based virtual server by using the command line interface

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

- o add lb vserver <name> (serviceType) <IPAddress> <port>
- o show lb vserver <name>

#### Example

```
> add lb vserver vssl SSL 10.102.29.133 443
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: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
Done
```

### To modify or remove an SSL-based virtual server by using the command line interface

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 [Customizing the SSL Configuration](#).

To remove an SSL virtual server, use the rm lb vserver command, which accepts only the <name> argument.

### To configure an SSL-based virtual server by using the configuration utility

Navigate to Traffic Management > Load Balancing > Virtual Servers, create a virtual server, and specify the protocol as SSL.

## Binding Services to the SSL-Based 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 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 NetScaler and the server. For details, see [Configuring SSL Offloading with End-to-End Encryption](#).

Note: The Load Balancing feature should be enabled on the NetScaler appliance before you bind services to the SSL based virtual server.

## To bind a service to a virtual server by using the command line interface

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>

### Example

```
> bind lb vserver vssl ssl1
Done
> show lb vserver vssl
vssl (10.102.29.133:443) - SSL Type: ADDRESS
State: DOWN[Certkey not bound]
Last state change was at Thu Nov 12 05:31:17 2009 (+485 ms)
Time since last state change: 0 days, 00:08:52.130
Effective State: DOWN
Client Idle Timeout: 180 sec
Down state flush: ENABLED
Disable Primary Vserver On Down : DISABLED
No. of Bound Services : 1 (Total) 1 (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) ssl1 (10.102.29.252: 80) - HTTP State: UP Weight: 1
Done
```

## To unbind a service from a virtual server by using the command line interface

At the command prompt, type the following command:

unbind lb vserver <name> <serviceName>

### Example

```
unbind lb vserver vssl ssl1
```

## To bind a service to 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 to bind a service to the virtual server.

## Adding or Updating a 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.

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 be prompted before the certificate expires.

Note: A certificate must be signed by using one of the following hash algorithms:

- o MD5
- o SHA-1
- o SHA-224
- o SHA-256
- o SHA-384
- o SHA-512

An MPX appliance supports certificates from 512-bits up to the following sizes:

- o 4096-bit server certificate on the virtual server
- o 4096-bit client certificate on the service
- o 4096-bit CA certificate (includes intermediate and root certificates)
- o 4096-bit certificate on the back end server
- o 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:

- o 4096-bit server certificate on the virtual server
- o 4096-bit client certificate on the service
- o 4096-bit CA certificate (includes intermediate and root certificates)
- o 2048-bit certificate on the back end server
- o 2048-bit client certificate (if client authentication is enabled on the virtual server)

## To add a certificate-key pair by using the command line interface

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

- o `add ssl certKey <certkeyName> -cert <string>[(-key <string> [-password]) | -fipsKey <string>] [-inform ( DER | PEM )] [<passplain>] [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <positive_integer>]]`
- o `show ssl certKey [<certkeyName>]`

### Example

```
> add ssl certKey sslckey -cert server_cert.pem -key server_key.pem -password ssl -expiryMon
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

## To update or remove a certificate-key pair by using the command line interface

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.

## To add or update a certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, and configure a certificate-key pair.

## Binding the Certificate-Key Pair to the SSL-Based 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 NetScaler 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 will accept 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 that have been issued by a trusted certificate authority.

## To bind an SSL certificate-key pair to a virtual server by using the command line interface

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>`

### 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES
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: Predefined Cipher Alias
Done
```

## To unbind an SSL certificate-key pair from a virtual server by using the command line interface

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>
```

### Example

```
unbind ssl vserver vssl -certkeyName sslkey
```

## To bind an SSL certificate-key pair to a virtual server by using the configuration utility

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.



## Configuring an SSL 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 NetScaler 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 Web sites 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 NetScaler 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. Based on the information provided by the client in the SNI extension, the NetScaler 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 same 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."

Note: On a NetScaler appliance, SNI is not supported with a Subject Alternative Name (SAN) extension certificate.

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.

### To bind multiple server certificates to a single SSL virtual server by using the command line interface

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>

To bind multiple server certificates to a transparent service by using the NetScaler command line, replace vserver with service and vservername with servicename in the above commands.

Note: The SSL service should be created with -clearTextPort 80 option.

#### Example

```
set ssl vserver v1 -snI ENABLED
bind ssl vserver v1 -certkeyName serverabc -SNICert
sh ssl vserver v1
Advanced SSL configuration for VServer v1:
â€¦
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: ENABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1)CertKey Name: servercert Server Certificate
1)CertKey Name: abccert Server Certificate for SNI
2)CertKey Name: xyzcert Server Certificate for SNI
3)CertKey Name: startcert Server Certificate for SNI
1)Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

### To bind multiple server certificates to a single SSL virtual server or transparent SSL service by using the configuration utility

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.

## Configuring a DTLS Virtual Server

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, EDH, DHE, ADH, EXP, and ECDHE ciphers are not supported on a DTLS virtual server.

## To create a DTLS configuration by using the command line

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>
```

The following steps are optional:

```
add dtlsProfile dtls1 -maxretryTime <positive_integer>
set ssl vserver <vserver_name> -dtlsProfileName <dtls_profile_name>
```

## To create a DTLS configuration by using the configuration utility

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.

## Example

The following example is for an end-to-end DTLS configuration:

```
> enable ns feature SSL LB
> add server s1 10.102.59.190
> add service svc1 s1 UDP 32000
> add lb vserver lb1 DTLS 10.102.59.244 443
> add ssl certKey servercert -cert server_cert.pem -key server_key.pem
> bind ssl vserver lb1 -certkeyname servercert
> bind lb vserver lb1 svc1

> sh lb vserver lb1
lb1 (10.102.59.244:443) - DTLS Type: ADDRESS
State: UP
Last state change was at Tue May 20 16:41:27 2014
Time since last state change: 0 days, 00:01:39.120
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) 1 (Active)
Configured Method: LEASTCONNECTION
Current Method: Round Robin, Reason: A new service is bound
Mode: IP
Persistence: NONE
L2Conn: OFF
Skip Persistency: None
IcmpResponse: PASSIVE
RHistate: PASSIVE
New Service Startup Request Rate: 0 PER_SECOND, Increment Interval: 0
TD: 0
```

```

Mac mode Retain Vlan: DISABLED
DBS_LB: DISABLED
Process Local: DISABLED

1 bound service:
1) svc1 (10.102.59.190: 32000) - UDP State: UP Weight: 1
Done
>
> sh ssl vservice lb1

Advanced SSL configuration for VServer lb1:
DH: 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
DTLSv1: ENABLED
Send Close-Notify: YES

DTLS profile name: nsdtls_default_profile

1 bound certificate:
1) CertKey Name: servercert Server Certificate

1 configured cipher:
1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done

> sh dtlsProfile nsdtls_default_profile
1) Name: nsdtls_default_profile
PMTU Discovery: DISABLED
Max Record Size: 1460 bytes
Max Retry Time: 3 sec
Hello Verify Request: DISABLED
Terminate Session: DISABLED
Max Packet Count: 120 bytes
Done

```

## 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

## 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.

### To create a DTLS profile by using the command line

- add ssl dtlsProfile <name>
- show ssl dtlsProfile<name>

#### 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
```

### To create a DTLS profile by using the configuration utility

Navigate to System > Profiles > DTLS Profiles and configure a new profile.

## Importing SSL Files from Remote Hosts

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.

### To import a certificate file from a remote host by using the command line

At the command prompt, type:

```
import ssl certFile [<name>] [<src>]
```

#### Example

```
import ssl certfile my-certfile http://www.example.com/file_1
> show ssl certfile
 Name : my-certfile
 URL : http://www.example.com/file_1
```

To remove a certificate file, use the `rm ssl certFile` command, which accepts only the `<name>` argument.

### To import a key file from a remote host by using the command line

At the command prompt, type:

```
import ssl keyFile [<name>] [<src>]
```

#### Example

```
import ssl keyfile my-keyfile http://www.example.com/key_file
> show ssl keyfile
 Name : my-keyfile
 URL : http://www.example.com/key_file
```

To remove a key file, use the `rm ssl keyFile` command, which accepts only the `<name>` argument.

### To import a CRL file from a remote host by using the command line

At the command prompt, type:

```
import ssl crlFile [<name>] [<src>]
```

#### 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
```

To remove a CRL file, use the `rm ssl crlFile` command, which accepts only the `<name>` argument.

### To import a DH file from a remote host by using the command line

At the command prompt, type:

```
import ssl dhFile [<name>] [<src>]
```

#### 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
```

To remove a DH file, use the `rm ssl dhFile` command, which accepts only the `<name>` argument.

## To import an SSL resource by using the configuration utility

Navigate to Traffic Management > SSL > Imports, and then select the appropriate tab.

## SSL Profiles

You can use an SSL profile to specify how a NetScaler ADC 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             |
| 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:

- Addâ€”Creates an SSL profile on the NetScaler ADC. Specify whether the profile is front end or backend. Front end is the default.

- o Setâ€”Modifies the settings of an existing profile.
- o Unsetâ€”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.
- o Removeâ€”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.
- o Showâ€”Displays all the profiles that are available on the NetScaler ADC . 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.

## To create an SSL profile by using the command line

- o To add an SSL profile, type: `add ssl profile <name> [-sslProfileType ( BackEnd | FrontEnd )]`
- o To modify an existing profile, type: `set ssl profile <name>`
- o To unset an existing profile, type: `unset ssl profile <name> [-dh] [-dhFile] [-dhCount] [-eRSA]â€|`
- o To unset an existing profile from an entity, type: `unset ssl vserver <vServerName> â€"sslProfile`
- o To remove an existing profile, type: `rm ssl profile <name>`
- o To display an existing profile, type: `sh ssl profile <name>`

## Examples

1. Adding a front end (default) profile:

```
> add sslprofile p1
Done
```

2. Adding a backend profile:

```
> add sslprofile p2 -sslprofileType backend -tls1 disabled
Done
```

3. Enabling settings on a backend profile:

```
> set sslprofile p2 -serverAuth ENABLED
Done
```

4. Enabling settings on a frontend profile:

```
> set sslprofile p1 -clientauth ENABLED -clientcert optional
Done
sh ssl profile p1
1) Configuration for Front-End SSL profile
Name: p1
DH: DISABLED
Ephemeral RSA: ENABLED Refresh Count: 0
Session Reuse: ENABLED Timeout: 120 seconds
Non FIPS Ciphers: DISABLED
Cipher Redirect: DISABLED
Client Auth: ENABLED Client Cert Required: Optional
SSL Redirect: DISABLED
SNI: DISABLED
SSLv3: ENABLED 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 NO
SSL quantum size: 8 kB
Strict CA checks: NO
Encryption trigger timeout 100 mS
Encryption trigger packet count: 45
Subject/Issuer Name Insertion Format: Unicode
Strict Host Header check for SNI enabled SSL sessions: NO
Done
```

5. Settings parameters to their default values:



```

> unset sslprofile p1 -clientauth -clientcert
Done
> sh ssl profile p1
1) Configuration for Front-End SSL profile
 Name: p1
 DH: DISABLED
 Ephemeral RSA: ENABLED Refresh Count: 0
 Session Reuse: ENABLED Timeout: 120 seconds
 Non FIPS Ciphers: DISABLED
 Cipher Redirect: DISABLED
 Client Auth: DISABLED
 SSL Redirect: DISABLED
 SNI: DISABLED
 SSLv3: ENABLED 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 NO
 SSL quantum size: 8 kB
 Strict CA checks: NO
 Encryption trigger timeout 100 mS
 Encryption trigger packet count: 45
 Subject/Issuer Name Insertion Format: Unicode
 Strict Host Header check for SNI enabled SSL sessions: NO
Done

```

#### 6. Deleting a profile:

```

> rm sslprofile p1
Done

```

#### 7. Binding a profile to a virtual server:

```

> set ssl vserver v1 -sslprofile p3
Done

```

#### 8. Unbinding a profile from a virtual server:

```

>unset ssl vserver v1 -sslprofile
Done

```

## To create an SSL profile by using the configuration utility

Navigate to System > Profiles, select the SSL Profiles tab, and create an SSL profile.

## Enabling Stricter Control on Client Certificate Validation

Note: This feature is supported from release 10.5 build 57.7 and later.

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.

### To enable stricter control on client certificate validation by using the command line

At the NetScaler command prompt, type:`set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

**To enable stricter control on client certificate validation by using the configuration utility**

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.

## Enabling 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.

### To enable stricter control on client certificate validation by using the command line

At the NetScaler command prompt, type: `set ssl profile <name> -ClientAuthUseBoundCAChain Enabled`

### To enable stricter control on client certificate validation by using the configuration utility

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.

## Managing Certificates

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.

Caution: Citrix recommends that you use certificates obtained from authorized CAs, such as VeriSign, for all your SSL transactions. Certificates generated on the NetScaler appliance should be used for testing purposes only, not in any live deployment.

To manage certificates, see the following sections:

- [Obtaining a Certificate from a Certificate Authority](#)
- [Importing Existing Certificates and Keys](#)
- [Generating a Self-Signed Certificate](#)
- [Adding a Group of SSL Certificates](#)
- [Displaying a Certificate Chain](#)
- [Generating a Server Test Certificate](#)
- [Modifying and Monitoring Certificates and Keys](#)
- [Using Global Site Certificates](#)
- [Converting the Format of SSL Certificates for Import or Export](#)

## Obtaining a Certificate from a Certificate Authority

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, you must 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.

## Creating 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 should 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, and the key is required for adding the certificate to the NetScaler.

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.

### To create an RSA private key by using the command line interface

At the command prompt, type the following command:

```
create ssl rsakey <keyFile> <bits> [-exponent (3 | F4)] [-keyform (DER | PEM)]
```

#### Example

```
> create ssl rsakey Key-RSA-1 2048 -exponent F4 -keyform PEM
```

### To create a DSA private key by using the command line interface

At the command prompt, type the following command:

```
create ssl dsakey <keyfile> <bits> [-keyform (DER | PEM)]
```

#### Example

```
> create ssl dsakey Key-DSA-1 2048 -keyform PEM
```

### To create an RSA private key by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Keys group, select Create RSA Key, and create an RSA key.

### To create a DSA private key by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Keys group, select Create DSA Key, and create a DSA key.

## Creating a Certificate Signing Request

The certificate signing request (CSR) is a collection of information, including the domain name, other important company details, and the private key to be used to create the certificate. To avoid generating an invalid certificate, make sure that the details you provide are accurate.

### To create a certificate signing request by using the command line interface

At the command prompt, type the following command:

```
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>]
```

#### Example

```
create ssl certreq csreq1 -keyfile ramp -keyform PEM -countryName US -stateName Florida -loc
```

## **To create a certificate signing request by using the configuration utility**

Navigate to Traffic Management > SSL and, in the SSL Certificates group, select Create Certificate Signing Request (CSR), and create a CSR.

## **Submitting the CSR to the CA**

Most CAs accept certificate submissions by email. The CA will return a valid certificate to the email address from which you submit the CSR.

## Importing Existing Certificates and Keys

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 NetScaler 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 [Converting the Format of SSL Certificates for Import or Export](#).

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 NetScaler command line 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-----
```

## To open the OpenSSL interface from the command line interface

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`.

## To open the ssl interface from the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select OpenSSL interface.

## To 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`:

- `rsa -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>`
- `dsa -in <PKCS#8 Key Filename> -des3 -out <encrypted Key Filename>`

## To convert a non-supported PKCS#8 key format to an unencrypted key format by using the OpenSSL interface

At the OpenSSL prompt, type the following commands, depending on whether the non-supported key format is of type `rsa` or `dsa`:

- `rsa -in <PKCS#8 Key Filename> -out <unencrypted Key Filename>`
- `dsa -in <PKCS#8 Key Filename> -out <unencrypted Key Filename>`

## 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.

## Generating a Self-Signed 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 itself, not by an actual CA, you should not use them in a production environment. If you attempt to use a self-signed certificate in a production environment, users will receive a "certificate invalid" warning each time the virtual server is accessed.

The NetScaler supports creation of the following types of certificates

- o Root-CA certificates
- o Intermediate-CA certificates
- o 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.

For details on how to create a private key and a CSR, see [Obtaining a Certificate from a Certificate Authority](#).

### To 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.

### To create a Root-CA certificate by using the command line interface

At the command prompt, type the following command:

```
create ssl cert <certFile> <reqFile> <certType> [-keyFile <input_filename>] [-keyform (DER | PEM)] [-days <positive_integer>]
```

#### Example

```
> create ssl cert cert1l csreq1 ROOT_CERT -keyFile
rsal -keyForm PEM -days 365
Done
```

### To create an Intermediate-CA certificate or end-user certificate by using the command line interface

At the command prompt, type the following command:

```
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>]
```

#### Example

```
> create ssl cert certsy csrl INTM_CERT -CAcert cert1
-CAkey rsakey1 -CAserial 23
Done
```

### To create a Root-CA certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the Getting Started group, select Root-CA Certificate Wizard, and configure a root CA certificate.

### To create an Intermediate-CA certificate or end-user certificate by using the configuration utility



Navigate to Traffic Management > SSL and, in the Getting Started group, select Intermediate-CA Certificate Wizard, and configure an intermediate CA certificate.

## Generating a Diffie-Hellman (DH) Key

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 a 2048-bit DH key may take a long time (up to 30 minutes).

### To generate a DH key by using the command line interface

At the command prompt, type the following command:

```
create ssl dhparam <dhFile> [<bits>] [-gen (2 | 5)]
```

#### Example

```
create ssl dhparam Key-DH-1 512 -gen 2
```

### To generate a DH key by using the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select Create Diffie-Hellman (DH) key, and generate a DH key.

## Adding a Group of SSL Certificates

If the server certificate is issued by an intermediate CA that is not recognized by standard web browsers as a trusted CA, 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.

## Adding and Linking a Certificate Set

Updated: 2014-06-17

Note: This feature is not supported on the NetScaler FIPS platform.

Instead of adding and linking individual certificates, you can now group a server certificate and up to nine intermediate certificates in a single file, and then 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 onNote: 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 already 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.

### To add a certificate set by using the command line interface

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

1. add ssl certKey <certkeyName> -cert <string> -key <string> -bundle (YES | NO)
2. show ssl certKey
3. show ssl certlink

### Example

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

> show ssl certkey

1) Name: bundle
Cert Path: /nsconfig/ssl/bundle.pem
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

2) Name: bundle_ic1
Cert Path: /nsconfig/ssl/bundle.pem_ic1
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

3) Name: bundle_ic2
Cert Path: /nsconfig/ssl/bundle.pem_ic2
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED

4) Name: bundle_ic3
Cert Path: /nsconfig/ssl/bundle.pem_ic3
Format: PEM
Status: Valid, Days to expiration:10415
Certificate Expiry Monitor: DISABLED
Done

> show ssl certlink

1) Cert Name: bundle CA Cert Name: bundle_ic1
2) Cert Name: bundle_ic1 CA Cert Name: bundle_ic2
3) Cert Name: bundle_ic2 CA Cert Name: bundle_ic3
Done

```

## To add a certificate set by using the configuration utility

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.

## Creating a Chain of Certificates

Updated: 2013-08-20

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 one of the certificates in the chain must be trusted by the client application. 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 NetScaler supports sending a maximum of 10 certificates in the chain of certificates sent to the client (one server certificate and nine CA certificates).

## To create a certificate chain by using the command line interface

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

### Example

```

> link ssl certkey siteAcertkey CAcertkey
Done

```

```
> show ssl certlink
```

```
linked certificate:
```

```
1) Cert Name: siteAcertkey CA Cert Name: CAcertkey
```

```
Done
```

### **To create a certificate chain by using the configuration utility**

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.

## Displaying 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 till 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 NetScaler 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.

## To view the certificate chain for the certificates present on the appliance by using the command line

At the command prompt, type:

```
show ssl certchain <cert_name>
```

### 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.

#### 1. 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
```

#### 2. 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
```

#### 3. 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
```

#### 4. 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
```

5. 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.
```

## Generating a Server Test Certificate

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. A server certificate is generally issued by a trusted CA and is sent out by the server to a client who uses it to authenticate the server.

For issuing a server test certificate, the appliance operates as a CA. 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. It should not be used in a production environment.

You can install the server test certificate on any virtual server that uses the SSL or the SSL\_TCP protocol.

### To generate a server test certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the SSL Certificates group, select Create and Install a Server Test Certificate.

## Modifying and Monitoring Certificates and Keys

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.

### Updating an Existing Server Certificate

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.

#### To update an existing certificate-key pair by using the command line interface

At the command prompt, type the following commands to update an existing certificate-key pair and verify the configuration:

- o `update ssl certkey <certkeyName> -cert <string> -key <string>`
- o `show ssl certKey <certkeyName>`

#### 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
```

#### To update an existing certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.

### Disabling Domain Checks

When an SSL certificate is replaced on the NetScaler, 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.

#### To disable the domain check for a certificate by using the command line interface

At the command prompt, type the following commands to disable the domain check and verify the configuration:

- o `update ssl certKey <certkeyName> -noDomainCheck`
- o `show ssl certKey <certkeyName>`

#### 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

```

## To disable the domain check for a certificate by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.
2. Select No Domain Check.

## Enabling the Expiry Monitor

An SSL certificate is valid for a specific period of time. 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 NetScaler appliance creates entries in the appliance's syslog and nsaudit logs when a certificate configured on the appliance is due to expire.

If you want to create SNMP alerts for certificate expiration, you must configure them separately.

For information about monitoring on the NetScaler appliance, see .

## To enable an expiry monitor for a certificate by using the command line interface

At the command prompt, type the following commands to enable an expiry monitor for a certificate and verify the configuration:

- o set ssl certKey <certkeyName> [-expiryMonitor ( ENABLED | DISABLED ) [-notificationPeriod <positive\_integer>]]
- o show ssl certKey <certkeyName>

### Example

```

> set ssl certKey sv -expiryMonitor ENABLED â€"notificationPeriod 60
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: ENABLED
 Expiry Notification period: 60 days
Done

```

## To enable an expiry monitor for a certificate by using the configuration utility

1. Navigate to Traffic Management > SSL > Certificates, select a certificate, and click Update.
2. Select Notify When Expires, and optionally specify a notification period.

## Using Global Site Certificates

A global site certificate is a special-purpose server certificate whose key length is greater than 128 bits. A global site certificate consists of a server certificate and an accompanying intermediate-CA certificate. You must import the global site certificate and its key from the server to the NetScaler appliance.

## How Global Site Certificates Work

Export versions of browsers use 40-bit encryption to initiate connections to SSL Web-servers. The server responds to connection requests by sending its certificate. The client and server then decide on an encryption strength based on the server certificate type:

- If the server certificate is a normal certificate and not a global site certificate, the export client and server complete the SSL handshake and uses 40-bit encryption for data transfer.
- If the server certificate is a global site certificate (and if the export client feature is supported by the browser) the export client automatically upgrades to 128-bit encryption for data transfer.

If the server certificate is a global site certificate, the server sends its certificate, along with the accompanying intermediate-CA certificate. The browser first validates the intermediate-CA certificate by using one of the Root-CA certificates that are normally included in web browsers. Upon successful validation of the intermediate-CA certificate, the browser uses the intermediate-CA certificate to validate the server certificate. Once the server is successfully validated, the browser renegotiates (upgrades) the SSL connection to 128-bit encryption.

With Microsoft's Server Gated Cryptography (SGC), if the Microsoft IIS server is configured with an SGC certificate, export clients that receive the certificate renegotiate to use 128-bit encryption.

## Importing a Global Site Certificate

To import a global site certificate, first export the certificate and server key from the Web server. Global site certificates are generally exported in some binary format, therefore, before importing the global site certificate, convert the certificate and key to the PEM format.

### To import a global site certificate

1. Using a text editor, copy the server certificate and the accompanying intermediate-CA certificate into two separate files.

The individual PEM encoded certificate will begin with the header -----BEGIN CERTIFICATE----- and end with the trailer -----END CERTIFICATE-----.

2. Use an SFTP client to transfer the server certificate, intermediate-CA certificate, and server-key to the NetScaler.
3. Use the following OpenSSL command to identify the server certificate and intermediate-CA certificate from the two separate files.

Note: You can launch the OpenSSL interface from the configuration utility. In the navigation pane, click SSL. In the details pane, under Tools, click Open SSL interface.

```
> openssl x509 -in >path of the CA cert file< -text
X509v3 Basic Constraints:
 CA:TRUE
X509v3 Key Usage:
 Certificate Sign, CRL Sign
Netscape Cert Type:
 SSL CA, S/MIME CA

> openssl x509 -in >path of the server certificate file< -text
X509v3 Basic Constraints:
 CA:FALSE
Netscape Cert Type:
 SSL Server
```

4. At the FreeBSD shell prompt, enter the following command:

```
openssl x509 -in cert.pem -text | more
```

Where **cert.pem** is one of the two certificate files.

Read the **Subject** field in the command output. For example,

```
Subject: C=US, ST=Oregon, L=Portland,
O=mycompany, Inc., OU=IT, CN=www.mycompany.com
```

If the CN field in the Subject matches the domain-name of your Web site, then this is the server certificate and the other certificate is the accompanying intermediate-CA certificate.

5. Use the server certificate and its private key) to create a certificate key pair on the NetScaler. For details on creating a certificate-key pair on the NetScaler, see [Adding a Certificate Key Pair](#).
6. Add the intermediate-CA certificate on the NetScaler. Use the server certificate you created in step 4 to sign this intermediate certificate. For details on creating an Intermediate-CA certificate on the NetScaler, see [Generating a Self Signed Certificate](#).

## Converting the Format of 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 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 configuration utility 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 NetScaler command line.

### To convert the format of a certificate by using the command line interface

At the command prompt, type the following command:

Convert `ssl pkcs12 <outfile> [-import [-pkcs12File <inputFilename>] [-des | -des3] [-export [-certFile <inputFilename>] [-keyFile <inputFilename>]]` 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
```

### To convert the format of a certificate by using the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select Import PKCS#12 or Export PKCS#12.

## Managing Certificate Revocation Lists

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. 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 NetScaler 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.

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.

To manage certificate revocation lists, see the following sections:

- [Creating a CRL on the NetScaler](#)
- [Adding an Existing CRL to the NetScaler](#)
- [Configuring CRL Refresh Parameters](#)
- [Synchronizing CRLs](#)
- [Performing Client Authentication by using a Certificate Revocation List](#)

## Creating a CRL on the NetScaler

Updated: 2013-09-04

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.

### To revoke a certificate or create a CRL by using the command line interface

At the command prompt, type the following command:

```
create ssl crl <CAcertFile> <CAkeyFile> <indexFile> (-revoke <input_filename> | -genCRL <output_filename>)
```

#### 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
```

### To revoke a certificate or create a CRL by using the configuration utility

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.

## Adding an Existing CRL to the NetScaler

Updated: 2013-09-05

Before you configure the CRL on the NetScaler appliance, make sure that the CRL file is stored locally on the NetScaler. 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.

## To add a CRL on the NetScaler by using the command line

At the command prompt, type the following commands to add a CRL on the NetScaler and verify the configuration:

- o add ssl crl <crlName> <crlPath> [-inform (DER | PEM)]
- o show ssl crl [<crlName>]

### 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=ww
 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
```

## To add a CRL on the NetScaler by using the configuration utility

Navigate to Traffic Management > SSL > CRL, and add a CRL.

## Configuring CRL Refresh Parameters

Updated: 2014-09-29

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 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.

## To configure CRL autorefresh by using the command line

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:

- o 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>]
- o show ssl crl [<crlName>]

### Example

```
Set CRL crl1 -refresh enabled -method ldap -inform DER -CAcert cal -server 10.102.192.192 -
set ssl crl crl1 -refresh enabled -method http -cacert cal -port 80 -time 00:10 -url http://
> sh crl

1) Name: crl1 Status: Valid, Days to expiration: 355
 CRL Path: /var/netScaler/ssl/crl1
 Format: PEM CAcert: cal
 Refresh: ENABLED Method: HTTP
 URL: http://10.102.192.192/crl/cal.crl Port:80
```

Done

## To configure CRL autorefresh using LDAP or HTTP by using the configuration utility

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.

To view the last update time, select the CRL, and click Details.

## Synchronizing CRLs

Updated: 2013-09-03

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 needs an automated mechanism to fetch the latest CRLs from the repository. You can configure the NetScaler to update CRLs automatically at a specified refresh interval.

The NetScaler 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.

## To synchronize CRL autorefresh by using the command line interface

At the command prompt, type the following command:

```
set ssl crl <crlName> [-interval <interval>] [-day <integer>] [-time <HH:MM>]
```

### Example

```
set ssl crl CRL-1 -refresh ENABLE -interval
MONTHLY -days 10 -time 12:00
```

## To synchronize CRL refresh by using the configuration utility

1. Navigate to Traffic Management > SSL > CRL.
2. Open a CRL, select enable CRL Auto Refresh, and specify the interval.

## Performing Client Authentication by using a Certificate Revocation List

Updated: 2013-09-03

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                                          |
| 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 command line interface

At the command prompt, type the following command:

```
bind ssl vserver <vServerName> -certkeyName <string> [(-CA -crlCheck (Mandatory | Optional
```

### Example

```
bind ssl vs v1 -certkeyName ca -CA -crlCheck mandatory
sh ssl vserver
> sh ssl vs 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: ENABLED Client Cert Required: Mandatory
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SNI: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
Push Encryption Trigger: Always
Send Close-Notify: YES
```

```
1) CertKey Name: ca CA Certificate CRLCheck: Mandatory CA_Name Sent
```

```
1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

## To configure CRL check by using the configuration utility

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.



## Monitoring Certificate Status with OCSP

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.

To monitor certificate status with OCSP, see the following sections:

- [NetScaler Implementation of OCSP](#)
- [OCSP Request Batching](#)
- [OCSP Response Caching](#)
- [Configuring an OCSP Responder](#)

## NetScaler Implementation of OCSP

OCSP validation on a NetScaler appliance begins when the appliance receives a client certificate during an SSL handshake. To validate the certificate, the NetScaler creates an OCSP request and forwards it to the OCSP responder. To do so, the NetScaler uses a locally configured URL. The transaction is in a suspended state until the NetScaler 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 NetScaler will allow the transaction or display an error, depending on whether the OCSP check was set to optional or mandatory, respectively.

The NetScaler 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 NetScaler receives a client certificate, it sends a request to the OCSP responder. To help avoid overloading the OCSP responder, the NetScaler 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 NetScaler caches the response locally for a predefined length of time. When a client certificate is received during an SSL handshake, the NetScaler 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 NetScaler sends a request to the OCSP responder and stores the response in its local cache for a configured length of time.

## Configuring an OCSP Responder

Updated: 2013-09-05

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.

### To add an OCSP responder by using the command line interface

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 )]`
- `bind ssl certKey [<certkeyName>] [-ocspResponder <string>] [-priority <positive_integer>]`
- `bind ssl vserver <vServerName>@ (-certkeyName <string> ( CA [-ocspCheck ( Mandatory | Optional )]))`
- `show ssl ocspResponder [<name>]`

### Example

```

add ssl ocsponder ocsponder1 -url "http:// www.myCA.org:80/ocsp/" -cache ENABLED -c
bind ssl certKey ca_cert -ocsponder ocsponder1 -priority 1
bind ssl vserver vs1 -certKeyName ca_cert -CA -ocspCheck Mandatory

sh ocsponder ocsponder1
1)Name: ocsponder1
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 ca_cert
Name: ca_cert Status: Valid, Days to expiration:8907
Version: 3
â€|
1) VServer name: vs1 CA Certificate
1) OCSponder name: ocsponder1 Priority: 1
Done

sh ssl vs vs1
Advanced SSL configuration for VServer vs1:
DH: DISABLED
â€|
1) CertKey Name: ca_cert CA Certificate OCSPCheck: Mandatory
1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done

```

## To modify an OCSP responder by using the command line interface

You cannot modify the responder name. All other parameters can be changed using the `set ssl ocsponder` command.

At the command prompt, type the following commands to set the parameters and verify the configuration:

- `set ssl ocsponder <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>] [-ocsponder <string>]`
- `bind ssl certKey [<certKeyName>] [-ocsponder <string>] [-priority <positive_integer>]`
- `show ssl ocsponder [<name>]`

## To configure an OCSP responder by using the configuration utility

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.

## Configuring Client Authentication

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 NetScaler to verify issuer signatures, the certificate of the CA that issued the client certificate must be installed on the NetScaler and bound to the virtual server that the client is transacting with.

If the certificate is valid, the NetScaler allows the client to access all secure resources. But if the certificate is invalid, the NetScaler drops the client request during the SSL handshake.

The NetScaler 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, 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 NetScaler.

Note: The NetScaler appliance supports a certificate-key pair size of 512 to 4096 bits. The certificate must be signed by using one of the following hash algorithms:

- o MD5
- o SHA-1
- o SHA-224
- o SHA-256
- o SHA-384
- o SHA-512

A NetScaler virtual appliance supports certificates of up to the following sizes:

- o 4096-bit server certificate on the virtual server
- o 4096-bit client certificate on the service
- o 4096-bit CA certificate
- o 2048-bit certificate on the physical server
- o 2048-bit client certificate (if client authentication is enabled on the virtual server)

To configure client authentication, see the following sections:

- o [Providing the Client Certificate](#)
- o [Enabling Client-Certificate-Based Authentication](#)
- o [Binding CA Certificates to the Virtual Server](#)

## Providing the Client Certificate

Before you configure client authentication, a valid client certificate must be 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, 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 [Converting SSL Certificates for Import or Export](#).

For instructions on how to generate a client certificate, see [Generating Self-Signed Certificates](#).

## Enabling 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 NetScaler requests the client certificate but proceeds with the SSL transaction even if the client presents an invalid certificate. If client authentication is mandatory, the NetScaler 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.

## To enable client-certificate-based authentication by using the command line interface

At the command prompt, type the following commands to enable the client-certificate-based authentication and verify the configuration:

- `set ssl vsrver <vServerName> [-clientAuth (ENABLED | DISABLED)] [-clientCert (MANDATORY | OPTIONAL)]`
- `show ssl vsrver <vServerName>`

### Example

```
> set ssl vsrver vssl -clientAuth ENABLED -clientCert Mandatory
Done
> show ssl vsrver 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```

```
1) CertKey Name: sslckey Server Certificate
1) Policy Name: client_cert_policy Priority: 0
1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
Done
```

## To enable client-certificate-based authentication by using the configuration utility

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.

## Binding 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 NetScaler 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 NetScaler 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.

For instructions on binding one or more certificates to the virtual server, see [Binding the Certificate-key Pair to the SSL Based Virtual Server](#).

For instructions on creating a chain of certificates, see [Creating a Chain of Certificates](#).

## Customizing the SSL Configuration

Once your basic SSL configuration is operational, you can customize some of the parameters that are specific to the certificates being used in SSL transactions. You can also enable and disable session reuse and client authentication, and you can configure redirect responses for cipher and SSLv2 protocol mismatches.

You can also customize SSL settings for two NetScaler appliances in a High Availability configuration, and you can synchronize settings, certificates and keys across those appliances.

These settings will depend on your network deployment and the type of clients you expect will connect to your servers.

To customize the SSL configuration, see the following sections:

- [Configuring Diffie-Hellman \(DH\) Parameters](#)
- [Configuring Ephemeral RSA](#)
- [Configuring Session Reuse](#)
- [Configuring Cipher Redirection](#)
- [Configuring SSLv2 Redirection](#)
- [Configuring SSL Protocol Settings](#)
- [Configuring Close-Notify](#)
- [Configuring ECDHE Ciphers](#)
- [Configuring a Common Name on an SSL Service or Service Group for Server Certificate Authentication](#)
- [Configuring Advanced SSL Settings](#)
- [Synchronizing Configuration Files in a High Availability Setup](#)
- [Managing Server Authentication](#)
- [Configuring User-Defined Cipher Groups on the NetScaler Appliance](#)

## Configuring Diffie-Hellman (DH) Parameters

If you are using ciphers on the NetScaler that require a DH key exchange to set up the SSL transaction, enable DH key exchange on the NetScaler and configure other settings based on your network.

To list the ciphers for which DH parameters must be set by using the NetScaler command line, 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 [Generating a Diffie-Hellman \(DH\) Key](#).

## To configure DH Parameters by using the command line interface

At the command prompt, type the following commands to configure DH parameters and verify the configuration:

- o `set ssl vsserver <vsServerName> -dh <Option> -dhCount <RefreshCountValue> -filepath <string>`
- o `show ssl vsserver <vServerName>`

### Example

```
> set ssl vsserver vs-server -dh ENABLED -dhFile /nsconfig/ssl/ns-server.cert -dhCount 1000
Done
> show ssl vsserver 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

## To configure DH Parameters by using the configuration utility

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.

## Configuring 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 at a later date 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.

## To configure Ephemeral RSA by using the command line interface

At the command prompt, type the following commands to configure ephemeral RSA and verify the configuration:

- o set ssl vservice <vServerName> -eRSA (enabled | disabled) -eRSACount <positive\_integer>
- o show ssl vservice <vServerName>

### Example

```
> set ssl vservice vs-server -eRSA ENABLED -eRSACount 1000
Done
> show ssl vservice vs-server

Advanced SSL configuration for VService 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED

1) Cipher Name: DEFAULT
Description: Predefined Cipher Alias
Done
```

## To configure Ephemeral RSA by using the configuration utility

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.



## Configuring 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.

### To configure session reuse by using the command line interface

At the command prompt, type the following commands to configure session reuse and verify the configuration:

- o `set ssl vsserver <vServerName> -sessReuse ( ENABLED | DISABLED ) -sessTimeout <positive_integer>`
- o `show ssl vsserver <vServerName>`

#### Example

```
> set ssl vsserver vs-ssl -sessreuse enabled -sesstimeout 600
Done
```

```
> show ssl vsserver 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```

```
1) CertKey Name: Auth-Cert-1 Server Certificate
```

```
1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
```

```
Done
```

### To configure session reuse by using the configuration utility

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.

## Configuring Cipher Redirection

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.

### To configure cipher redirection by using the command line interface

At the command prompt, type the following commands to configure cipher redirection and verify the configuration:

- o set ssl vserver <vServerName> -cipherRedirect < ENABLED | DISABLED> -cipherURL < URL>
- o show ssl vserver <vServerName>

#### 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
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```

```
1) CertKey Name: Auth-Cert-1 Server Certificate
```

```
1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
```

```
Done
```

### To configure cipher redirection by using the configuration utility

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.

## Configuring SSLv2 Redirection

For an SSL transaction to be initiated, and for successful completion of the SSL handshake, the server and the client should agree on an SSL protocol that both of them support. If the SSL protocol version supported by the client is not acceptable to the server, the server does not go ahead with the transaction, and an error message is displayed.

You can configure the server to display a precise error message (user-configured or internally generated) advising the client on the next action to be taken. Configuring the server to display this message requires that you set up SSLv2 redirection.

### To configure SSLv2 redirection by using the command line interface

At the command prompt, type the following commands to configure SSLv2 redirection and verify the configuration:

- o set ssl vserver <vServerName> [-ssl2Redirect ( ENABLED | DISABLED ) [-ssl2URL <URL>]]
- o show ssl vserver <vServerName>

#### Example

```
> set ssl vserver vs-ssl -ssl2Redirect ENABLED -ssl2URL http://ssl2URL
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: ENABLED Redirect URL: http://ssl2URL
ClearText Port: 0
Client Auth: DISABLED
SSL Redirect: DISABLED
Non FIPS Ciphers: DISABLED
SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
```

```
1) CertKey Name: Auth-Cert-1 Server Certificate
```

```
1) Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
```

```
Done
```

### To configure SSLv2 redirection by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select SSLv2 Redirect, and specify a URL.

## Configuring SSL Protocol Settings

The NetScaler appliance supports the SSLv2, 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 will connect to the appliance.

Note: Support for TLS protocol versions 1.1 and 1.2 is not available on a FIPS appliance or on a NetScaler virtual 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.

Note: TLS\_FALLBACK\_SCSV is supported from release 10.5 build 57.7 and later.

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.

## To configure SSL protocol support by using the command line interface

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>`

### 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
Session Reuse: ENABLED
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
Push Encryption Trigger: Always
Send Close-Notify: YES

Refresh Count: 0
Timeout: 120 seconds

1) 1 bound certificate:
 CertKey Name: mycert Server Certificate

1) 1 configured cipher:
 Cipher Name: DEFAULT
 Description: Predefined Cipher Alias
Done
```

## To configure SSL protocol support by using the configuration Utility

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.

## Configuring 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 [Configuring Advanced SSL Settings](#).

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.

### To configure close-notify at the entity level by using the command line interface

At the command prompt, type any of the following commands to configure close-notify and verify the configuration:

1. To configure close-notify at the virtual server level, type:
  - o `set ssl vserver <vServerName> -sendCloseNotify ( YES | NO )`
  - o `show ssl vserver <vServerName>`
2. To configure close-notify at the service level, type:
  - o `set ssl service <serviceName> -sendCloseNotify ( YES | NO )`
  - o `show ssl service <serviceName>`
3. To configure close-notify at the service group level, type:
  - o `set ssl serviceGroup <serviceGroupName> -sendCloseNotify ( YES | NO )`
  - o `show ssl serviceGroup <serviceGroupName>`

#### Example

```
> set ssl vserver sslvsrv -sendCloseNotify YES
Done
```

### To configure close-notify at the entity level by using the configuration utility

1. Navigate to Traffic Management > Load Balancing > Virtual Servers, and open a virtual server.
2. In the SSL Parameters section, select Send Close-Notify.

## Configuring ECDHE Ciphers

The Citrix NetScaler VPX, MPX, and SDX appliances support the ECDHE cipher group in release 10.5 build 53.9 or later. 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 a complete list of ciphers supported by the NetScaler appliance, see [Ciphers Supported by the NetScaler Appliance](#).

The following table lists the ciphers supported on VPX instances and MPX appliances.

| Cipher Suite                       | VPX | MPX |
|------------------------------------|-----|-----|
| TLS1-ECDHE-RSA-RC4-SHA             | YES | YES |
| TLS1-ECDHE-RSA-DES-CBC3-SHA        | YES | YES |
| TLS1-ECDHE-RSA-AES128-SHA          | YES | YES |
| TLS1-ECDHE-RSA-AES256-SHA          | YES | YES |
| TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 | NO  | YES |
| TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 | NO  | YES |
| TLS1.2-ECDHE-RSA-AES-128-SHA256    | NO  | YES |
| TLS1.2-ECDHE-RSA-AES-256-SHA384    | NO  | YES |

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:

- o P\_256
- o P\_384
- o P\_224
- o P\_521

Note:

- o On the MPX platform, ECC curves 224 and 521 are not supported with the TLS1.2 protocol.
- o 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.

## To unbind ECC curves and bind an ECC curve to an SSL virtual server by using the command line

At the command prompt, type:

- o unbind ssl vserver <vServerName> -eccCurveName ALL
- o bind ssl vserver <vServerName> -eccCurveName <eccCurveName>

### 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    TL  
Push Encryption Trigger: Always  
Send Close-Notify: YES

ECC Curve: P\_224

- 1)            Cipher Name: DEFAULT  
             Description: Predefined Cipher Alias
- Done

## Configuring a Common Name on an SSL Service or Service Group 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. 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.

### To configure common-name verification for an SSL service or service group by using the command line interface

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:
  - o set ssl service <serviceName> -commonName <string> -serverAuth ENABLED
  - o show ssl service <serviceName>
2. To configure common name in a service group, type:
  - o set ssl serviceGroup <serviceGroupName> -commonName <string> -serverAuth ENABLED
  - o show ssl serviceGroup <serviceGroupName>

#### Example

```
> set ssl service svc1 -commonName xyz.com -serverAuth ENABLED
Done
> show ssl service svc1
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
```

### To configure common-name verification for an SSL service or service group by using the configuration utility

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.



## Configuring Advanced SSL Settings

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.

## To configure advanced SSL settings by using the command line interface

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`

### Example

```
> set ssl parameter -quantumSize 8 -crlMemorySizeMB 256 -strictCACHecks no -sslTriggerTimeout 100 -sendCloseNotify no -encryptTriggerPktCount 45 -denySSLReneg no -insertionEncoding unicode -ocspCacheSize 10 -pushFlag 3 -dropReqWithNoHostHeader YES -pushEncTriggerTimeout 100
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 PUSH encryption trigger tim
```

## To configure advanced SSL settings by using the configuration utility

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 <positive_integer>` command.

## To configure PUSH flag-based encryption by using the command line interface

At the command prompt, type the following commands to configure PUSH flag-based encryption and verify the configuration:

- o set ssl vserver <vServerName> [-pushEncTrigger <pushEncTrigger>]
- o show ssl vserver

### Example

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       |
| Push Encryption Trigger: Always | TLSv1: ENABLED       |

### To configure PUSH flag-based encryption by using the configuration utility

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.

## Synchronizing Configuration Files in a High Availability Setup

In a high availability (HA) set up, the primary NetScaler appliance in the HA pair automatically synchronizes with the secondary appliance in the pair. In the synchronization process, the secondary copies the primary's /nsconfig/ssl/ directory, which is the default location for storing the certificates and keys for SSL transactions. Synchronization occurs at one-minute intervals and every time a new file is added to the directory.

### To synchronize files in a high availability setup by using the command line interface

At the command prompt, type the following command:

```
sync HA files [<Mode>]
```

#### Example

```
sync HA files SSL
```

### To synchronize files in a high availability setup by using the configuration utility

Navigate to Traffic Management > SSL and, in the Tools group, select Start SSL certificate, key file synchronization for HA.

## Managing Server Authentication

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 NetScaler 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. When binding the certificate, you must specify the bind as CA option.

### To enable (or disable) server certificate authentication by using the command line interface

At the command prompt, type the following commands to enable server certificate authentication and verify the configuration:

- o `set ssl service <serviceName> -serverAuth ( ENABLED | DISABLED )`
- o `show ssl service <serviceName>`

#### Example

```
> set ssl service ssl-service-1 -serverAuth ENABLED
Done
> 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
```

### To enable (or disable) server certificate authentication by using the configuration utility

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.

### To bind the CA certificate to the service by using the command line interface

At the command prompt, type the following commands to bind the CA certificate to the service and verify the configuration:

- o `bind ssl service <serviceName> -certkeyName <string> -CA`
- o `show ssl service <serviceName>`

#### Example

```
> bind ssl service ssl-service-1 -certkeyName samplecertkey -CA
Done
> 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

## Configuring User-Defined Cipher Groups on the NetScaler Appliance

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
```

2. Bind the AES cipher group to the service

```
bind ssl service <Service name> -cipherName AE
```

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
```

Note: The free NetScaler virtual appliance supports only the DH cipher group.

## To configure a user-defined cipher group by using the command line interface

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>

### Example

```
> add ssl cipher test
Done
> bind ssl cipher test -cipherName SSLv2
Done
> show ssl cipher test
1) Cipher Name: SSL2-RC2-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC2(128) Mac=MD5
2) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
3) Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
4) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
5) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
6) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
7) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done
```

## To unbind ciphers from a cipher group by using the command line interface

At the command prompt, type the following commands to unbind ciphers from a user-defined cipher group, and verify the settings:

- o show ssl cipher <cipherGroupName>
- o unbind ssl cipher <cipherGroupName> -cipherName <string>
- o show ssl cipher <cipherGroupName>

### Example

```
> show ssl cipher test
1) Cipher Name: SSL2-RC2-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC2(128) Mac=MD5
2) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
3) Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
4) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
5) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
6) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
7) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done

> unbind ssl cipher test -cipherName SSL2-RC2-CBC-MD5

> show ssl cipher test
1) Cipher Name: SSL2-RC4-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(128) Mac=MD5
2) Cipher Name: SSL2-DES-CBC3-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=3DES(168) Mac=MD5
3) Cipher Name: SSL2-DES-CBC-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=DES(56) Mac=MD5
4) Cipher Name: SSL2-RC4-64-MD5
Description: SSLv2 Kx=RSA Au=RSA Enc=RC4(64) Mac=MD5
5) Cipher Name: SSL2-EXP-RC4-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC4(40) Mac=MD5 Export
6) Cipher Name: SSL2-EXP-RC2-CBC-MD5
Description: SSLv2 Kx=RSA(512) Au=RSA Enc=RC2(40) Mac=MD5 Export
Done
```

## To remove a cipher group by using the command line interface

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:

- o rm ssl cipher <userDefCipherGroupName> [<cipherName> ...]
- o show ssl cipher <cipherGroupName>

### Example

```
> rm ssl cipher test
Done

> sh ssl cipher test
ERROR: No such resource [cipherGroupName, test]
```

## To configure a user-defined cipher group by using the configuration utility

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 command line interface

At the command prompt, type one of the following:

- o bind ssl vserver <vServerName> -cipherName <string>
- o bind ssl service <serviceName> -cipherName <string>
- o bind ssl serviceGroup <serviceGroupName> -cipherName <string>

### Examples

```
> bind ssl vserver ssl_vserver_test -cipherName test
Done
bind ssl service nshttps -cipherName test
Done
> bind ssl servicegroup ssl_svc -cipherName test
Done
```

## To bind a cipher group to an SSL virtual server, service, or service group by using the configuration utility

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.



## Configuring SSL Actions and Policies

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 NetScaler command line 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.

To configure SSL actions and policies, see the following sections:

- [Configuring User-Defined Actions for SSL Policies](#)
- [Configuring SSL Policies](#)
- [Configuring an SSL Default Syntax Policy](#)
- [Configuring Built-in Actions for SSL Default Syntax Policies](#)
- [Configuring SSL Policy Labels](#)
- [Configuring Per-Directory Client Authentication](#)
- [Configuring Support for Outlook Web Access](#)
- [Configuring SSL-Based Header Insertion](#)
- [Binding SSL Policies to a Virtual Server](#)
- [Binding SSL Policies Globally](#)

## Configuring User-Defined Actions for SSL Policies

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 [Configuring Built-in SSL Actions](#).

### To configure an SSL action by using the command line interface

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)
- show ssl action [<name>]

#### Example

```
> add ssl action Action-SSL-ClientCert -clientCert ENABLED -certHeader "X-Client-Cert"
Done
> show ssl action Action-SSL-ClientCert
1) Name: Action-SSL-ClientCert
 Data Insertion Action:
 Cert Header: ENABLED Cert Tag: X-Client-Cert
Done
```

### To configure an SSL action by using the configuration utility

Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.

## Configuring SSL Policies

Policies on the NetScaler 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 NetScaler, 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.

The NetScaler SSL feature supports both classic policies and default syntax policies . For a complete description of classic and default syntax expressions, how they work, and how to configure them manually, see .

**Note:** Users who are not experienced in configuring policies at the NetScaler command line will usually find using the configuration utility considerably easier.

## Configuring an SSL Default Syntax Policy

An SSL default syntax 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. SSL policies use the default expression syntax or the classic expression syntax. For information about expressions and how to configure them, see .

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.

### To configure an SSL default syntax policy by using the command line interface

At the command prompt, type:

```
add ssl policy <name> -rule <expression> -Action <string> [-undefAction <string>] [-comment <string>]
```

### To configure an SSL default syntax policy by using the configuration utility

Navigate to Traffic Management > SSL > Policies and, on the Policies tab, click Add.

## Configuring Built-in Actions for SSL Default Syntax Policies

Unless you need only the built-in actions in your policies, you have to create the actions before creating the policies, so that 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:

- CLIENTAUTHâ€”Perform client certificate authentication.
- NOCLIENTAUTHâ€”Do not perform client certificate authentication.

The built-in data actions are:

- RESETâ€”Close the connection by sending a RST packet to the client.
- DROPâ€”Drop all packets from the client. The connection remains open until the client closes it.
- NOOPâ€”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. For more information about user-defined actions, see [Configuring User-Defined SSL Actions](#).

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
```

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
```

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
```

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
```

## Configuring SSL Policy Labels

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 NetScaler command line, 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.

### To create an SSL policy label and bind policies to the label by using the command line interface

At the command prompt, type:

- `add ssl policylabel <labelName> -type ( CONTROL | DATA )`
- `bind ssl policylabel <labelName> <policyName> <priority> [<gotoPriorityExpression>] [-invoke (<labelType> <labelName>)]`

#### 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
```

### To configure an SSL policy label and bind policies to the label by using the configuration utility

Navigate to Traffic Management > SSL > Policy Labels, and configure an SSL policy label.

## Configuring Per-Directory Client Authentication

If you create an action specifying client-side authentication on a per-directory basis, a client identified by a policy associated with the action is not authenticated as part of the initial SSL handshake. Instead, authentication is carried out every time the client wants to access a specific directory on the web server.

For example, if you have multiple divisions in the company, where each division has a folder in which all its files are stored, and you want to know the identity of each client that tries to access files from a particular directory, such as the finance directory, you can enable per-directory client authentication for that directory.

To enable per-directory client authentication, first configure client authentication as an SSL action, and then create a policy that identifies the directory that you want to monitor. When you create the policy, specify your client-authentication action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

### To create an SSL action and a policy to enable client authentication by using the command line interface

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

- add ssl action <name> [-clientAuth ( DOCLIENTAUTH | NOCLIENTAUTH )]
- show ssl action [<name>]
- add ssl policy <name> -rule <expression> [-action <string>] [-undefAction <string>] [-comment <string>]
- show ssl policy [<name>]

#### Example

```
> add ssl action ssl-action-1 -clientAuth DOCLIENTAUTH
Done
> show ssl action ssl-action-1
1) Name: ssl-action-1
 Client Authentication Action: DOCLIENTAUTH
 Hits: 0
 Undef Hits: 0
 Action Reference Count: 1
Done
> add ssl policy ssl-pol-1 -rule 'REQ.HTTP.METHOD==GET' -reqaction ssl-action-1
> sh ssl policy ssl-pol-1
 Name: ssl-pol-1
 Rule: REQ.HTTP.METHOD == GET
 Action: ssl-action-1
 UndefAction: Use Global
 Hits: 0
 Undef Hits: 0
Done
```

### To create an SSL action to enable client authentication by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.
2. In the Client Authentication list, select Enabled.

### To create and bind an SSL policy to enable client authentication by using the configuration utility

1. Navigate to Traffic Management > SSL and, on the Polices tab, click Add.
2. Navigate to Traffic Management > Load Balancing > Virtual Servers and open an SSL virtual server.

In Advanced Settings, select SSL Policy, and bind the policy to the virtual server.

## Configuring Support for Outlook Web Access

If your SSL configuration is offloading SSL transactions from an Outlook Web Access (OWA) server, you must insert a special header field, FRONT-END-HTTPS: ON, in all HTTP requests directed to the OWA servers. This is required for the OWA servers to generate URL links as https:// instead of http://.

When you enable support for OWA on the NetScaler, the header is automatically inserted into the specified HTTP traffic, and you do not need to configure a specific header insertion. Use SSL policies to identify all traffic directed to the OWA server.

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

To enable OWA support, first configure OWA support as an SSL action, and then create a policy that identifies the virtual servers or services for which you want to enable OWA support. When you create the policy, specify your OWA support action as the action associated with the policy. Then, bind the policy to the SSL virtual server that will receive the SSL traffic.

### To create an SSL action and a policy to enable OWA support by using the command line interface

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 | DISABLED )
- show ssl action [<name>]
- add ssl policy <name> -rule <expression> [-action <string>] [-undefAction <string>] [-comment <string>]
- show ssl policy [<name>]

#### Example

```
> add ssl action ssl-action-2 -OWASupport ENABLED
Done
> show ssl action ssl-action-2
1) Name: ssl-action-2
 Type: Data Insertion
 OWA Support: ENABLED

 Hits: 0
 Undef Hits: 0
 Action Reference Count: 1

Done
> add ssl policy ssl-pol -rule 'REQ.HTTP.METHOD == GET' -reqaction ssl-action-2
Done
> sh ssl policy ssl-pol

 Name: ssl-pol
 Rule: REQ.HTTP.METHOD == GET
 Action: ssl-action-2
 UndefAction: Use Global
 Hits: 0
 Undef Hits: 0

Done
```

### To create an SSL action to enable OWA support by using the configuration utility

1. Navigate to Traffic Management > SSL > Policies and, on the Actions tab, click Add.
  2. In the Outlook Web Access list, select Enabled.
- Note: Outlook Web Access support is applicable only for SSL virtual server based configurations and transparent SSL service based configurations and not for SSL configurations with back-end encryption.

### To create and bind an SSL policy to enable OWA support by using the configuration utility

Navigate to Traffic Management > SSL > Policies, and add a policy.

In the Action list, select the action that you created earlier. Specify an undefined action, and an expression.



## Configuring SSL-Based Header Insertion

Because the NetScaler appliance offloads all SSL-related processing from the servers, the servers receive only HTTP traffic. In some circumstances, the server needs certain SSL information. For example, security audits of recent SSL transactions require the client subject name (contained in an X509 certificate) to be logged on the server.

Such data can be sent to the server by inserting it into the HTTP header as a name-value pair. You can insert the entire client certificate, if required, or only the specific fields from the certificate, such as the subject, serial number, issuer, certificate hash, SSL session ID, cipher suite, or the not-before or not-after date used to determine certificate validity.

You can enable SSL-based insertion for HTTP-based SSL virtual servers and services only. You cannot apply it to TCP-based SSL virtual servers and services. Also, client authentication must be enabled on the SSL virtual server, because the inserted values are taken from the client certificate that is presented to the virtual server for authentication.

To configure SSL-based header insertion, first create an SSL action for each specific set of information to be inserted, and then create policies that identify the connections for which you want to insert the information. As you create each policy, specify the action that you want associated with the policy. Then, bind the policies to the SSL virtual servers that will receive the SSL traffic.

The following example uses default syntax policies. In the following example, a control policy (ctrlpol) is created to perform client authentication if a request is received for the URL /testsite/file5.html. A data policy (datapol) is created to perform an action (act1) if client authentication is successful, and an SSL action (act1) is added to insert the certificate details and issuer's name in the request before forwarding the request. For other URLs, client authentication is disabled. The policies are then bound to an SSL virtual server (ssl\_vserver) that receives the SSL traffic.

## Command-line example of configuring SSL-based header insertion

### Example

```
> add ssl action act1 -clientCert ENABLED -certHeader mycert -clientcertissuer ENABLED -cert
> add ssl policy datapol -rule HTTP.REQ.URL.EQ("/testsite/file5.html") -action act1
> add ssl policy ctrlpol -rule HTTP.REQ.URL.EQ("/testsite/file5.html") -action CLIENTAUTH
> bind ssl vserver ssl_vserver -policyName ctrlpol -priority 1
> bind ssl vserver ssl_vserver -policyName datapol -priority 1
Done
```

## To configure SSL-based header insertion by using the configuration utility

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, set the following parameters:
  - o Name\*
  - o Client Certificate
  - o Certificate Tag
  - o Client Certificate Issuer
  - o Issuer Tag
- \* A required parameter
4. Click Create, and then click Close.
5. On the tab, click Add to create a control policy.
6. In the Create SSL Policy dialog box, set the following parameters:
  - o Name\*
  - o Expression
  - o Request Action
- \* A required parameter
7. Click Create, and then click Close.
8. Create a data policy by repeating steps 5 through 7.
9. In the navigation pane, expand SSL Offload, and then click Virtual Servers.
10. In the details pane, from the list of virtual servers, select the virtual server to which you want to bind the SSL policies, and then click Open.
11. In the Configure Virtual Server (SSL Offload) dialog box, click SSL Settings, and then click SSL Policies.
12. In the Bind/Unbind SSL Policies dialog box, click Insert Policy. Under Policy Name, select the policy that you created in steps 5 through 7.
13. Click OK, and then click Close. A message appears in the status bar, stating that the policy has been bound successfully.

14. Repeat steps 12 and 13 and select the policy that you created in step 8.

## Binding SSL Policies to a Virtual Server

The SSL policies that are configured on the NetScaler appliance need to be bound to a virtual server that intercepts traffic directed to the virtual server. 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.

You can also bind SSL policies globally or to custom bind points on the NetScaler appliance. For more information about binding policies on the appliance, see .

### To bind an SSL policy to a virtual server by using the command line interface

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

- `bind ssl vsrver <vServerName> -policyName <string> [-priority <positive_integer>]`
- `show ssl vsrver <vServerName>`

#### Example

```
> bind ssl vsrver vs-server -policyName ssl-policy-1 -priority 10
Done
> show ssl vsrver 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
```

### To bind an SSL policy to a virtual server by using the configuration utility

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.

## Binding SSL Policies Globally

Globally bound policies are evaluated after all policies bound to services, virtual servers, or other NetScaler bind points are evaluated.

### To globally bind an SSL policy by using the command line interface

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`

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
```

### To bind a global SSL policy by using the configuration utility

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.

## Use Case 1: Configuring SSL Offloading with End-to-End Encryption

A simple SSL offloading setup terminates SSL traffic (HTTPS), decrypts the SSL records, and forwards the clear text (HTTP) traffic to the back-end web servers. However, the clear text traffic is vulnerable to being spoofed, read, stolen, or compromised by individuals who succeed in gaining access to the back-end network devices or web servers.

You can, therefore, configure SSL offloading with end-to-end security by re-encrypting the clear text data and using secure SSL sessions to communicate with the back-end Web servers.

Additionally, you can configure the back-end SSL transactions so that the NetScaler appliance uses SSL session multiplexing to reuse existing SSL sessions with the back-end web servers, thus avoiding CPU-intensive key exchange (full handshake) operations. This reduces the overall number of SSL sessions on the server, and therefore accelerates the SSL transaction while maintaining end-to-end security.

To configure SSL Offloading with end-to-end encryption, add SSL based services that represent secure servers with which the NetScaler appliance will carry out end-to-end encryption. Then create an SSL based virtual server, and create and bind a valid certificate-key pair to the virtual server. Bind the SSL services to the virtual server to complete the configuration.

For details on adding SSL based services, see [Configuring Services](#).

For details on adding an SSL virtual server, see [Configuring an SSL Based Virtual Server](#).

For details on creating a certificate-key pair, see [Adding a Certificate-Key Pair](#).

For details on binding a certificate-key pair to a virtual server, see [Binding the Certificate Key Pair to the SSL Based Virtual Server](#).

For details on binding services to a virtual server, see [Binding Services to the SSL Based Virtual Server](#).

### Example

Create two SSL based services, Service-SSL-1 and Service-SSL-2, with IP addresses 10.102.20.30 and 10.102.20.31 and both using port 443.

Then create an SSL based virtual server, Vserver-SSL-2 with an IP address of 10.102.10.20.

Next, create a certificate-key pair, CertKey-1 and bind it to the virtual server.

Bind the SSL services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Offloading with End-to-End Encryption Example

| Entity                   | Name          | Value        |
|--------------------------|---------------|--------------|
| SSL Service              | Service-SSL-1 | 10.102.20.30 |
| Â                        | Service-SSL-2 | 10.102.20.31 |
| SSL Based Virtual Server | Vserver-SSL-2 | 10.102.10.20 |
| Certificate - Key Pair   | Certkey-1     | Â            |

## Use Case 2: Configuring Transparent SSL Acceleration

Note: You need to enable L2 mode on the NetScaler appliance for transparent SSL acceleration to work.

Transparent SSL acceleration is useful for running multiple applications on a secure server with the same public IP, and also for SSL acceleration without using an additional public IP.

In a transparent SSL acceleration setup, the NetScaler appliance is transparent to the client, because the IP address at which the appliance receives requests is the same as the Web server's IP address.

The NetScaler offloads SSL traffic processing from the Web server and sends either clear text or encrypted traffic (depending on the configuration) to the web server. All other traffic is transparent to the NetScaler and is bridged to the Web server. Therefore, other applications running on the server are unaffected.

There are three modes of transparent SSL acceleration available on the NetScaler:

- Service-based transparent access, where the service type can be SSL or SSL\_TCP.
- Virtual server-based transparent access with a wildcard IP address (\*:443).
- SSL VIP-based transparent access with end-to-end encryption.

Note: An SSL\_TCP service is used for non-HTTPS services (for example SMTPS and IMAPS).

### Service-based Transparent SSL Acceleration

To enable transparent SSL acceleration using the SSL service mode, configure an SSL or an SSL\_TCP service with the IP address of the actual back-end Web server. Instead of a virtual server intercepting SSL traffic and passing it on to the service, the traffic is now directly passed on to the service, which decrypts the SSL traffic and sends clear text data to the back-end server.

The service-based mode allows you to configure individual services with a different certificate, or with a different clear text port. Also, you can also select individual services for SSL acceleration.

You can apply service-based transparent SSL acceleration to data that uses different protocols, by setting the clear text port of the SSL service to the port on which the data transfer between the SSL service and the back-end server occurs.

To configure service-based transparent SSL acceleration, first enable both the SSL and the load balancing features. Then create an SSL based service and configure its clear text port. After the service is created, create and bind a certificate-key pair to this service.

For details on configuring the clear text port for an SSL based service, see ["Configuring Advanced SSL Settings."](#)

For details on creating a certificate-key pair and binding a certificate-key pair to a service, see ["Adding a Certificate-Key Pair."](#)

#### Example

Enable SSL offloading and load balancing.

Create an SSL based service, Service-SSL-1 with the IP address 10.102.20.30 using port 443 and configure its clear text port.

Next, create a certificate-key pair, CertKey-1 and bind it to the SSL service.

Table 1. Entities in the Service-based Transparent SSL Acceleration

| Entity                 | Name          | Value     |
|------------------------|---------------|-----------|
| SSL Service            | Service-SSL-1 | 102.20.30 |
| Certificate - Key Pair | Certkey-1     | Â         |

### Virtual Server-based Acceleration with a Wildcard IP Address (\*:443)

You can use an SSL virtual server in the wildcard IP address mode if when you want to enable SSL acceleration for multiple servers that host the secure content of a Web site. In this mode, a single-digital certificate is enough for the entire secure Web site, instead of one certificate per virtual server. This results in significant cost savings on SSL certificates and renewals. The wildcard IP address mode also enables centralized certificate management.

To configure global transparent SSL acceleration on the NetScaler appliance, create a \*:443 virtual server, which is a virtual server that accepts any IP address associated with port 443. Then, bind a valid certificate to this virtual server, and also bind all services to which the virtual server is to transfer. Such a virtual server can use the SSL protocol for HTTP-based data or the SSL\_TCP protocol for non-HTTP-based data.

**To configure virtual server-based acceleration with a wildcard IP address**

- 1. Enable SSL, as described in "Enabling SSL Processing."
  - 2. Enable load balancing, as described in "Load Balancing."
  - 3. Add an SSL based virtual server (see "Configuring an SSL-Based Virtual Server" for the basic settings), and set the clearTextPort parameter (described in "Configuring Advanced SSL Settings)."
  - 4. Add a certificate-key pair, as described in "Adding a Certificate-Key Pair."
- Note: The wildcard server will automatically learn the servers configured on the NetScaler, so you do not need to configure services for a wildcard virtual server.

**Example**

After enabling SSL offloading and load balancing, create an SSL based wildcard virtual server with IP address set to \* and port number 443, and configure its clear text port (optional).

If you specify the clear text port, decrypted data will be sent to the backend server on that particular port. Otherwise, encrypted data will be sent to port 443.

Next, create an SSL certificate key pair, CertKey-1 and bind it to the SSL virtual server.

Table 2. Entities in the Virtual Server-based Acceleration with a Wildcard IP Address Example

| Entity                   | Name                 | IP Address | Port |
|--------------------------|----------------------|------------|------|
| SSL Based Virtual Server | Vserver-SSL-Wildcard | *          | 443  |
| Certificate - Key Pair   | Certkey-1            | Â          | Â    |

**SSL VIP-based Transparent Access with End-To-End Encryption**

You can use an SSL virtual server for transparent access with end-to-end encryption if you have no clear text port specified. In such a configuration, the NetScaler terminates and offloads all SSL processing, initiates a secure SSL session, and sends the encrypted data, instead of clear text data, to the web servers on the port that is configured on the wildcard virtual server.

Note: In this case, the SSL acceleration feature runs at the back-end, using the default configuration, with all 34 ciphers available.

To configure SSL VIP based transparent access with end-to-end encryption, Follow instructions for Configuring a Virtual Server-based Acceleration with a Wildcard IP Address (\*:443), but do not configure a clear text port on the virtual server.

## Use Case 3: Configuring SSL Acceleration with HTTP on the Front End and SSL on the Back End

In certain deployments, you might be concerned about network vulnerabilities between the NetScaler appliance and the backend servers, or you might need complete end-to-end security and interaction with certain devices that can communicate only in clear text (for example, caching devices).

In such cases, you can set up an HTTP virtual server that receives data from clients that connect to it at the front end and hands the data off to a secure service, which securely transfers the data to the web server.

To implement this type of configuration, you configure an HTTP virtual server on the NetScaler and bind SSL based services to the virtual server. The NetScaler receives HTTP requests from the client on the configured HTTP virtual server, encrypts the data, and sends the encrypted data to the web servers in a secure SSL session.

To configure SSL acceleration with HTTP on the front-end and SSL on the back-end, first enable the load balancing and SSL features on the NetScaler. Then, add SSL based services that represent secure servers to which the NetScaler appliance will send encrypted data. Finally, add an HTTP based virtual server and bind the SSL services to this virtual server.

### Example

Enable load balancing and SSL acceleration on the NetScaler.

After enabling load balancing and SSL acceleration, create two SSL based services, Service-SSL-1 and Service-SSL-2, with IP addresses 10.102.20.30 and 10.102.20.31, and both using port 443.

Then create an HTTP based virtual server, Vserver-HTTP-1, with an IP address of 10.102.10.20.

Bind the SSL services to the virtual server to complete the configuration.

Table 1. Entities in the SSL Acceleration with HTTP on the Front End and SSL on the Back End Example

| Entity                    | Name           | Value        |
|---------------------------|----------------|--------------|
| SSL Service               | Service-SSL-1  | 10.102.20.30 |
| Â                         | Service-SSL-2  | 10.102.20.31 |
| HTTP Based Virtual Server | Vserver-HTTP-1 | 10.102.10.20 |



## Use Case 4: SSL Offloading with Other TCP Protocols

In addition to the secure HTTP (HTTPS) protocol, NetScaler appliances support SSL acceleration for other TCP-based secure protocols. However, only simple requests and response-based TCP application protocols are supported. Applications such as FTPS, that insert the server's IP address and port information in their payloads, are not currently supported.

Note: The STARTTLS feature for SMTP is currently not supported.

The NetScaler supports SSL acceleration for Other TCP protocols with and without end-to-end encryption.

To configure SSL offloading with Other TCP protocols, create a virtual server of type `SSL_TCP`, bind a certificate-key pair and TCP based services to the virtual server, and configure SSL actions and policies based on the type of traffic expected and the acceleration to be provided.

Follow the instructions in [Configuring SSL Offloading](#), but create an `SSL_TCP` virtual server instead of an SSL virtual server, and configure TCP services instead of HTTP services.

### SSL\_TCP Based Offloading with End-to-End Encryption

To configure `SSL_TCP`-based offloading with end-to-end encryption, both the virtual server that intercepts secure traffic and the services that it forwards the traffic to must be of type `SSL_TCP`.

Configure `SSL_TCP`-based offloading as described in [Configuring SSL Offloading with End-to-End Encryption](#), but create an `SSL_TCP` virtual server instead of an SSL virtual server.

### Backend Encryption for TCP Based Data

Some deployments might require the NetScaler appliance to encrypt TCP data received as clear text and send the data securely to the back end servers.

To provide SSL acceleration with back-end encryption for clear text TCP traffic arriving from the client, create a TCP based virtual server and bind it to `SSL_TCP` based services.

To configure end-to-end encryption for TCP-based data, follow the procedure described in [Configuring the SSL feature with HTTP on the Front-End and SSL on the Back-End](#), but create a TCP virtual server instead of an HTTP virtual server.

## Use Case 5: Configuring SSL Bridging

An SSL bridge configured on the NetScaler appliance enables the appliance to bridge all secure traffic between the SSL client and the SSL server. The appliance does not offload or accelerate the bridged traffic, nor does it perform encryption or decryption. Only load balancing is done by the appliance. The SSL server must handle all SSL-related processing. Features such as content switching, SureConnect, and cache redirection do not work, because the traffic passing through the appliance is encrypted.

Because the appliance does not carry out any SSL processing in an SSL bridging setup, there is no need for SSL certificates.

Citrix recommends that you use this configuration only if an acceleration unit (for example, a PCI-based SSL accelerator card) is installed in the web server to handle the SSL processing overhead.

Before you configure SSL bridging, first enable SSL and load balancing on the appliance. Then, create SSL\_Bridge services and bind them to an SSL\_Bridge virtual server. Configure the load balancing feature to maintain server persistency for secure requests.

### Example

After enabling SSL and load balancing, create two servers, s1 and s2. Create two SSL\_Bridge services, sc1 and src2. Create an SSL\_Bridge virtual server and bind the SSL\_Bridge services to the virtual server to complete the configuration. At the command line, type:

```
enable ns feature SSL LB
add server s1 10.102.1.101
add server s2 10.102.1.102
add service src1 s1 SSL_BRIDGE 443
add service src2 s2 SSL_BRIDGE 443
add lb vserver ssl_bridge_vip SSL_BRIDGE 10.102.1.200 443
bind lb vserver ssl_bridge_vip src1
bind lb vserver ssl_bridge_vip src2
```

## Use Case 6: Configuring SSL Monitoring when Client Authentication is Enabled on the Backend Service

Consider a scenario in which you need to load balance servers that require SSL client certificates to validate clients. For this deployment, you need to create an SSL service on the NetScaler appliance, add an HTTPS monitor, add a certificate-key pair, bind this certificate-key pair to the SSL service, and then bind the https monitor to this service. You can use this https monitor to perform health checks on the backend services.

### To configure SSL monitoring with client certificate

1. Open an SSH connection to the appliance by using an SSH client, such as PuTTY.
2. Log on the appliance by using the administrator credentials.
3. Add an SSL service. At the command prompt, type:

```
add service <name> <serverName> <serviceType> <port>
```

4. Add an https monitor. At the command prompt, type:

```
add lb monitor <name> <type>
```

5. Add the certificate-key pair that is going to be used as the client cert for that SSL service. At the command prompt, type:

```
add ssl certKey <certkeyName> -cert <string> -key <string>
```

6. Bind this certkey to the SSL service. At the command prompt, type:

```
bind ssl service <serviceName> -certkeyName <string>
```

7. Bind the https monitor to the SSL service. At the command prompt, type:

```
bind lb monitor <monitorName> <serviceName>
```

Now, when the appliance tries to probe the backend service on which client authentication is enabled, the backend service will request a certificate as part of the SSL handshake. When the appliance returns the certificate-key bound in step 6 above, the monitor probe will succeed.

### Example

```
add service svc_k 10.102.145.30 SSL 443
add lb monitor sslmon HTTP -respCode 200 -httpRequest "GET /testsite/file5.html" -secure YE
add ssl certKey ctest -cert client_rsa_2048.pem -key client_rsa_2048.ky
bind ssl service svc_k -certkeyName ctest
bind lb monitor sslmon svc_k
> show service svc_k
 svc_k (10.102.145.30:443) - SSL
 State: UP
 Last state change was at Tue Jan 10 13:12:24 2012
 Time since last state change: 0 days, 00:09:37.890
 Server Name: 10.102.145.30
 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
 Appflow logging: ENABLED

1) Monitor Name: sslmon
 State: UP Weight: 1
 Probes: 1318 Failed [Total: 738 Current: 0]
 Last response: Success - HTTP response code 200 received.
 Response Time: 0.799 millisec
Done
>
```

```
> show ssl service svc_k
 Advanced SSL configuration for Back-end SSL Service svc_k:
 DH: DISABLED
 Ephemeral RSA: DISABLED
 Session Reuse: ENABLED Timeout: 300 seconds
 Cipher Redirect: DISABLED
 SSLv2 Redirect: DISABLED
 Server Auth: DISABLED
 SSL Redirect: DISABLED
 Non FIPS Ciphers: DISABLED
 SNI: DISABLED
 SSLv2: DISABLED SSLv3: ENABLED TLSv1: ENABLED
1) CertKey Name: ctest Client Certificate

1) Cipher Name: ALL
 Description: Predefined Cipher Alias
Done
```

## Use Case 7: Configuring a Secure Content Switching Server

An SSL-based content switching virtual server first decrypts the secure data and then redirects the data to appropriately configured servers as determined by the type of content and the configured content switching policies. The packets sent to the server have a mapped IP address as the source IP address.

The following example shows the steps to configure two address-based virtual servers to perform load balancing on the HTTP services. One virtual server, Vserver-LB-HTML, load balances the dynamic content (cgi, asp), and the other, Vserver-LB-Image, load balances the static content (gif, jpeg). The load-balancing method used is the default, LEASTCONNECTION. A content switching SSL virtual server, Vserver-CS-SSL, is then configured to perform SSL acceleration and switching of HTTPS requests on the basis of configured content switching policies.

### Example

```
> enable ns feature lb cs ssl
> add lb vserver Vserver-LB-HTML http 10.1.1.2 80
> add lb vserver Vserver-LB-Image http 10.1.1.3 80
> add service s1 10.1.1.4 http 80
> add service s2 10.1.1.5 http 80
> add service s3 10.1.1.6 http 80
> add service s4 10.1.1.7 http 80
> bind lb vserver Vserver-LB-HTML s1
> bind lb vserver Vserver-LB-HTML s2
> bind lb vserver Vserver-LB-Image s3
> bind lb vserver Vserver-LB-Image s4
> add cs vserver Vserver-CS-SSL ssl 10.1.1.1 443
> add cs policy pol1 -url "*.cgi"
> add cs policy pol2 -url "*.asp"
> add cs policy pol3 -url "*.gif"
> add cs policy pol4 -url "*.jpeg"
> bind cs vserver Vserver-CS-SSL -policyName pol1 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol2 Vserver-LB-HTML
> bind cs vserver Vserver-CS-SSL -policyName pol3 Vserver-LB-Image
> bind cs vserver Vserver-CS-SSL -policyName pol4 Vserver-LB-Image
> add certkey mykey -cert /nsconfig/ssl/ns-root.cert -key /nsconfig/ssl/ns-root.key
> bind certkey Vserver-CS-SSL mykey
>
> show cs vserver Vserver-CS-SSL
^ ^ ^ ^ ^ ^ ^ Vserver-CS-SSL (10.1.1.1:443) - SSL Type: CONTENT
^ ^ ^ ^ ^ ^ ^ State: UP
^ ^ ^ ^ ^ ^ ^ Last state change was at Tue Jul 13 02:11:37 2010
^ ^ ^ ^ ^ ^ ^ Time since last state change: 0 days, 00:02:12.440
^ ^ ^ ^ ^ ^ ^ Client Idle Timeout: 180 sec
^ ^ ^ ^ ^ ^ ^ Down state flush: ENABLED
^ ^ ^ ^ ^ ^ ^ Disable Primary Vserver On Down : 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
```

## Ciphers Supported by the NetScaler Appliance

Your NetScaler appliance ships with a predefined set of cipher groups. Table 1 lists the ciphers that are part of the DEFAULT cipher group and are therefore bound by default to an SSL virtual server. Table 2 lists the other ciphers currently supported by the NetScaler appliance. 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 more information about creating a user-defined cipher group, see [Configuring User-Defined Cipher Groups on the NetScaler Appliance](#).

Note:

- On all NetScaler appliances, connection between a NetScaler service and the back end server can be established using TLS protocol version 1.0 or SSLv3 protocol only.
- Support for TLS versions 1.1 and 1.2 is only available on the front end (between client and virtual server) on the following appliances:
  - NetScaler MPX appliances.
  - NetScaler VPX appliances from build 57.7.
  - NetScaler SDX appliances on an instance-by-instance basis. To support TLS protocol versions 1.1 and 1.2 on an SDX appliance, you must assign at least one SSL chip to the instance when you provision it.
- Support for TLS protocol versions 1.1 and 1.2 is not available on a FIPS appliance.

Table 1. Ciphers That the NetScaler Appliance Supports by Default

| Cipher Suite         | Protocol | Key Exchange Algorithm | Authentication Algorithm | Encryption Algorithm (Key Size) | Message Authentication Code Algorithm |
|----------------------|----------|------------------------|--------------------------|---------------------------------|---------------------------------------|
| SSL3-RC4-MD5         | SSLv3    | RSA                    | RSA                      | RC4(128)                        | MD5                                   |
|                      | TLSv1    |                        |                          |                                 |                                       |
|                      | TLSv1.1  |                        |                          |                                 |                                       |
|                      | TLSv1.2  |                        |                          |                                 |                                       |
| SSL3-RC4-SHA         | SSLv3    | RSA                    | RSA                      | RC4(128)                        | SHA                                   |
|                      | TLSv1    |                        |                          |                                 |                                       |
|                      | TLSv1.1  |                        |                          |                                 |                                       |
|                      | TLSv1.2  |                        |                          |                                 |                                       |
| SSL3-DES-CBC3-SHA    | SSLv3    | RSA                    | RSA                      | 3DES(168)                       | SHA                                   |
|                      | TLSv1    |                        |                          |                                 |                                       |
|                      | TLSv1.1  |                        |                          |                                 |                                       |
|                      | TLSv1.2  |                        |                          |                                 |                                       |
| TLS1-AES-256-CBC-SHA | SSLv3    | RSA                    | RSA                      | AES(256)                        | SHA                                   |
|                      | TLSv1    |                        |                          |                                 |                                       |
|                      | TLSv1.1  |                        |                          |                                 |                                       |
|                      | TLSv1.2  |                        |                          |                                 |                                       |
| TLS1-AES-128-CBC-SHA | SSLv3    | RSA                    | RSA                      | AES(128)                        | SHA                                   |
|                      | TLSv1    |                        |                          |                                 |                                       |

|                              |                                      |    |     |           |     |
|------------------------------|--------------------------------------|----|-----|-----------|-----|
|                              | TLSv1.1                              |    |     |           |     |
|                              | TLSv1.2                              |    |     |           |     |
| SSL3-EDH-DSS-DES-CBC3-SHA    | SSLv3<br>TLSv1                       | DH | DSS | 3DES(168) | SHA |
| TLS1-DHE-DSS-RC4-SHA         | TLSv1                                | DH | DSS | RC4(128)  | SHA |
| TLS1-DHE-DSS-AES-256-CBC-SHA | SSLv3<br>TLSv1                       | DH | DSS | AES(256)  | SHA |
| TLS1-DHE-DSS-AES-128-CBC-SHA | SSLv3<br>TLSv1                       | DH | DSS | AES(128)  | SHA |
| SSL3-EDH-RSA-DES-CBC3-SHA    | SSLv3<br>TLSv1<br>TLSv1.1<br>TLSv1.2 | DH | RSA | 3DES(168) | SHA |
| TLS1-DHE-RSA-AES-256-CBC-SHA | SSLv3<br>TLSv1<br>TLSv1.1<br>TLSv1.2 | DH | RSA | AES(256)  | SHA |
| TLS1-DHE-RSA-AES-128-CBC-SHA | SSLv3<br>TLSv1<br>TLSv1.1<br>TLSv1.2 | DH | RSA | AES(128)  | SHA |

Table 2. Additional Ciphers Supported by the NetScaler Appliance

| Cipher Suite         | Protocol                  | Key Exchange Algorithm | Authentication Algorithm | Encryption Algorithm (Key Size) | Message Authentication Code Algorithm |
|----------------------|---------------------------|------------------------|--------------------------|---------------------------------|---------------------------------------|
| SSL3-DES-CBC-SHA     | SSLv3<br>TLSv1<br>TLSv1.1 | RSA                    | RSA                      | DES(56)                         | SHA                                   |
| TLS1-EXP1024-RC4-SHA | TLSv1                     | RSA (1024)             | RSA                      | RC4(56)                         | SHA                                   |
| SSL3-EXP-RC4-MD5     | SSLv3<br>TLSv1            | RSA(512)               | RSA                      | RC4(40)                         | MD5                                   |

|                                   |                           |               |      |           |    |
|-----------------------------------|---------------------------|---------------|------|-----------|----|
| SSL3-EXP-DES-CBC-SHA              | SSLv3<br>TLSv1            | RSA(512)      | RSA  | DES(40)   | St |
| SSL3-EXP-RC2-CBC-MD5              | SSLv3<br>TLSv1            | RSA(512)      | RSA  | RC2(40)   | MI |
| SSL2-RC4-MD5                      | SSLv2                     | RSA           | RSA  | RC4(128)  | MI |
| SSL2-DES-CBC3-MD5                 | SSLv2                     | RSA           | RSA  | 3DES(168) | MI |
| SSL2-RC2-CBC-MD5                  | SSLv2                     | RSA           | RSA  | RC2(128)  | MI |
| SSL2-DES-CBC-MD5                  | SSLv2                     | RSA           | RSA  | DES(56)   | MI |
| SSL2-RC4-64-MD5                   | SSLv2                     | RSA           | RSA  | RC4(64)   | MI |
| SSL2-EXP-RC4-MD5                  | SSLv2                     | RSA(512)      | RSA  | RC4(40)   | MI |
| SSL3-EDH-DSS-DES-CBC-SHA          | SSLv3<br>TLSv1            | DH            | DSS  | DES(56)   | St |
| TLS1-EXP1024-DHE-DSS-DES-CBC- SHA | TLSv1                     | DH(1024)      | DSS  | DES(56)   | St |
| TLS1-EXP1024-DHE-DSS-RC4- SHA     | TLSv1                     | DH(1024)      | DSS  | RC4(56)   | St |
| SSL3-EXP-EDH-DSS-DES-CBC-SHA      | SSLv3<br>TLSv1            | DH(512)       | DSS  | DES(40)   | St |
| SSL3-EDH-RSA-DES-CBC-SHA          | SSLv3<br>TLSv1<br>TLSv1.1 | DH            | RSA  | DES(56)   | St |
| SSL3-EXP-EDH-RSA-DES-CBC-SHA      | SSLv3<br>TLSv1            | DH(512)       | RSA  | DES(40)   | DE |
| TLS1-EXP1024-RC4-MD5              | TLSv1                     | RSA<br>(1024) | RSA  | RC4(56)   | MI |
| TLS1-EXP1024-RC2-CBC-MD5          | TLSv1                     | RSA<br>(1024) | RSA  | RC2(56)   | MI |
| SSL2-EXP-RC2-CBC-MD5              | SSLv2                     | RSA(512)      | RSA  | RC2(40)   | MI |
| SSL3-ADH-RC4-MD5                  | SSLv3<br>TLSv1            | DH            | None | RC4(128)  | MI |



|                             |                             |         |      |           |    |
|-----------------------------|-----------------------------|---------|------|-----------|----|
|                             | TLSv1.1                     |         |      |           |    |
| SSL3-ADH-DES-CBC-SHA        | SSLv3<br>TLSv1<br>TLSv1.1   | DH      | None | DES(56)   | St |
| SSL3-ADH-DES-CBC3-SHA       | SSLv3<br>TLSv1<br>TLSv1.1   | DH      | None | 3DES(168) | St |
| TLS1-ADH-AES-128-CBC-SHA    | SSLv3<br>TLSv1<br>TLSv1.1   | DH      | None | AES(128)  | St |
| TLS1-ADH-AES-256-CBC-SHA    | SSLv3<br>TLSv1<br>TLSv1.1   | DH      | None | AES(256)  | St |
| SSL3-EXP-ADH-RC4-MD5        | SSLv3<br>TLSv1              | DH(512) | None | RC4(40)   | MI |
| SSL3-EXP-ADH-DES-CBC-SHA    | SSLv3<br>TLSv1              | DH(512) | None | DES(40)   | St |
| TLS1-ECDHE-RSA-RC4-SHA      | TLSv1<br>TLSv1.1<br>TLSv1.2 | ECC-DHE | RSA  | RC4(128)  | St |
| TLS1-ECDHE-RSA-DES-CBC3-SHA | TLSv1<br>TLSv1.1<br>TLSv1.2 | ECC-DHE | RSA  | 3DES(168) | St |
| TLS1-ECDHE-RSA-AES128-SHA   | TLSv1<br>TLSv1.1<br>TLSv1.2 | ECC-DHE | RSA  | AES(128)  | St |
| TLS1-ECDHE-RSA-AES256-SHA   | TLSv1<br>TLSv1.1<br>TLSv1.2 | ECC-DHE | RSA  | AES(256)  | St |

Table 3. Additional Ciphers Supported with TLS Protocol Version 1.2 on Front End Entities from Release 10.5 Build 53.9

| Cipher Suite | Protocol | Key Exchange | Authentication Algorithm | Encryption Algorithm | Message Authentication |
|--------------|----------|--------------|--------------------------|----------------------|------------------------|
|--------------|----------|--------------|--------------------------|----------------------|------------------------|

|                                    |         | Algorithm |     | (Key Size)    | Code (MAC) Algorithm |
|------------------------------------|---------|-----------|-----|---------------|----------------------|
| TLS1.2-AES128-GCM-SHA256           | TLSv1.2 | RSA       | RSA | AES-GCM (128) | SHA-256              |
| TLS1.2-AES256-GCM-SHA384           | TLSv1.2 | RSA       | RSA | AES-GCM (256) | SHA-384              |
| TLS1.2-DHE-RSA-AES128-GCM-SHA256   | TLSv1.2 | DH        | RSA | AES-GCM (128) | SHA-256              |
| TLS1.2-DHE-RSA-AES256-GCM-SHA384   | TLSv1.2 | DH        | RSA | AES-GCM (256) | SHA-384              |
| TLS1.2-ECDHE-RSA-AES128-GCM-SHA256 | TLSv1.2 | ECC-DHE   | RSA | AES-GCM (128) | SHA-256              |
| TLS1.2-ECDHE-RSA-AES256-GCM-SHA384 | TLSv1.2 | ECC-DHE   | RSA | AES-GCM (256) | SHA-384              |
| TLS1.2-ECDHE-RSA-AES-128-SHA256    | TLSv1.2 | ECC-DHE   | RSA | AES(128)      | SHA-256              |
| TLS1.2-ECDHE-RSA-AES-256-SHA384    | TLSv1.2 | ECC-DHE   | RSA | AES(256)      | SHA-384              |
| TLS1.2-AES-256-SHA256              | TLSv1.2 | RSA       | RSA | AES(256)      | SHA-256              |
| TLS1.2-AES-128-SHA256              | TLSv1.2 | RSA       | RSA | AES(128)      | SHA-256              |
| TLS1.2-DHE-RSA-AES-128-SHA256      | TLSv1.2 | DH        | RSA | AES(128)      | SHA-256              |
| TLS1.2-DHE-RSA-AES-256-SHA256      | TLSv1.2 | DH        | RSA | AES(256)      | SHA-256              |

Note: From release 10.5 build 53.9, ECDHE, AES-GCM, and SHA2 ciphers are part of the default group. ECDHE/DHE cipher suites must be used to achieve perfect forward secrecy (PFS). However, AES-GCM and SHA2 ciphers are supported on the front-end SSL entities only.

From build 58.11, the following ECDHE ciphers are also supported on the back end on the NetScaler MPX platforms:

- TLS1-ECDHE-RSA-RC4-SHA
- TLS1-ECDHE-RSA-DES-CBC3-SHA
- TLS1-ECDHE-RSA-AES128-SHA
- TLS1-ECDHE-RSA-AES256-SHA

On a NetScaler platform that does not have N3 chips and is configured to negotiate EDH ciphers by using TLS version 1.0 with a DH key of 2048 bits, the SSL handshake fails in either of the following scenarios:

- o Client authentication is enabled and the appliance receives a client certificate of 4096 bits.
- o End-to-end encryption is configured and the appliance receives a server certificate of 4096 bits.

Use the show ns hardware command to find out if your appliance has N3 chips.

### Example

```
> sh 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
```

## FIPS

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.

This section includes the following details:

- [Configuring the HSM](#)
- [Creating and Transferring FIPS Keys](#)
- [Configuring FIPS Appliances in a High Availability Setup](#)
- [Updating the Firmware to Version 2.2 on a FIPS Card](#)
- [Resetting a Locked HSM](#)
- [FIPS Approved Algorithms and Ciphers](#)

## Configuring the HSM

Before you can configure the HSM of your NetScaler FIPS appliance, you must complete the initial hardware configuration. For more information, see .

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.

## To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

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:

1. `show ssl fips`
2. `reset ssl fips`
3. `reboot`
4. `set ssl fips -initHSM Level-2 <newSOpassword> <oldSOpassword> <userPassword> [-hsmLabel <string>]`
5. `save ns config`
6. `reboot`
7. `show ssl fips`

### 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 sol2345 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: In release 10.5.e build 55.8007\_e and later, 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
```

## To configure the HSM on an MPX 9700/10500/12500/15500 FIPS appliances by using the configuration utility

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:
  - o Security Officer (SO) Password\*â€”new SO password
  - o Old SO Password\*â€”old SO password
  - o User Password\*â€”user password
  - o Levelâ€”initHSM (Currently set to Level2 and cannot be changed)
  - o HSM Labelâ€”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.

## Creating and Transferring 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.

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.

### Creating a FIPS Key

Updated: 2013-08-20

Before creating a FIPS key, make sure that the HSM is configured.

#### To create a FIPS key by using the configuration utility

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\*â€”fipsKeyName
  - Modulus\*â€”modulus
  - Exponent\*â€”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.

#### To create a FIPS key by using the command line interface

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>]

#### Example

```
create fipskey Key-FIPS-1 -modulus 2048 -exponent 3
show ssl fipsKey Key-FIPS-1
FIPS Key Name: Key-FIPS-1 Modulus: 2048 Public Exponent: 3 (Hex: 0x3)
```

### Exporting a FIPS Key

Updated: 2013-08-20

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.

#### To export a FIPS key by using the command line interface

At the command prompt, type:  
export ssl fipsKey <fipsKeyName> -key <string>

#### Example

```
export fipskey Key-FIPS-1 -key Key-FIPS-1.key
```

#### To export a FIPS key by using the configuration utility

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:
  - o FIPS Key Name\*â€”fipsKeyName
  - o File Name\*â€”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.

## Importing an Existing FIPS Key

Updated: 2013-11-22

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.

### To import a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

At the command prompt, type the following commands to import a FIPS key and verify the settings:

- o import ssl fipskey <fipsKeyName> -key <string> -inform SIM -exponent (F4 | 3)
- o show ssl fipskey <fipsKeyName>

#### 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)
```

### To import a FIPS key by using the configuration utility

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:
  - o FIPS Key Name\*
  - o Key File Name\*â€”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.
  - o 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.

## Importing External Keys

Updated: 2015-02-09

In addition to transferring FIPS keys that are created within the NetScaler applianceâ€™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.

### Importing an external key as a FIPS key on the MPX 9700/10500/12500/15500 FIPS appliances by using the command line interface

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>
```

### To import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the command line interface

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:
  - o `convert ssl pkcs12 <output file> -import -pkcs12File <input .pfx file name> -password <password>`
3. At the command prompt, type the following commands to import the external key as a FIPS key and verify the settings:
  - o `import ssl fipsKey <fipsKeyName> -key <string> -inform PEM`
  - o `show ssl fipskey<fipsKeyName>`

### 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)
```

Note: The modulus is incorrectly displayed as zero in the above example. The discrepancy does not affect SSL functionality.

### To import an external key as a FIPS key to an MPX 9700/10500/12500/15500 FIPS appliance by using the configuration utility

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:
      - o Output File Name\*
      - o PKCS12 File Name\*â€”Specify the .pfx file name.
      - o Import Password\*
      - o 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:
    - o FIPS Key Name\*
    - o Key File Name\*â€”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.



## Configuring FIPS Appliances in a High Availability Setup

You can configure two appliances in a high availability (HA) pair as FIPS appliances. For information about configuring an HA setup, see .

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 inconsistency that requires that the appliance be restarted. In addition, if you use the CLI, the `create ssl fipskey` command is not propagated to the secondary node. When you execute the command with the same input values for modulus size and exponent on two different FIPS appliances, the keys generated are not identical. You have to 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 should 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.

### To configure FIPS appliances in a high availability setup by using the command line interface

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>
```

4. Copy this <certFile> file to appliance B, in the /nconfig/ssl folder.
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>
```

8. Copy this <targetSecret> file to appliance A.
9. **On appliance A**, enable appliance A as the source appliance. At the command prompt, type:

```
enable ssl fipsSIMSource <targetSecret> <sourceSecret>
```

10. Copy this <sourceSecret> file to appliance B.
11. **On appliance B**, enable appliance B as the target appliance. At the command prompt, type:

```
enable ssl fipsSIMtarget <keyVector> <sourceSecret>
```

12. **On appliance A**, create a FIPS key, as described in [Creating a FIPS Key](#).
13. Export the FIPS key to the appliance's hard disk, as described in [Exporting 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 [Importing an Existing FIPS Key](#).

### To configure FIPS appliances in a high availability setup by using the configuration utility

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, in the Certificate File Name text box, type the file name, with the path to the location at which the FIPS certificate should 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 should 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 [Creating a FIPS Key](#). The FIPS key is automatically transferred from the primary to the secondary.

## Example

In the following example, source.cert is the certificate on the source appliance, stored in the default directory, /nsconfig/ssl. This certificate must be transferred to the same location (/nsconfig/ssl) on the target appliance. The file target.secret is created on the target appliance and copied to the source appliance. The file source.secret is created on the source appliance and copied to the target appliance.

### On the source appliance

```
init fipsSIMsource /nsconfig/ssl/source.cert
```

### On the target appliance

```
init fipsSIMtarget /nsconfig/ssl/source.cert /nsconfig/ssl/target.key /nsconfig/ssl/target.
```

### On the source appliance

```
enable fipsSIMsource /nsconfig/ssl/target.secret /nsconfig/ssl/source.secret
```

### On the target appliance

```
enable fipsSIMtarget /nsconfig/ssl/target.key /nsconfig/ssl/source.secret
```

### On the source appliance

```
create ssl fipskey fips1 -modulus 2048 -exponent f4
export fipskey fips1 -key /nsconfig/ssl/fips1.key
```

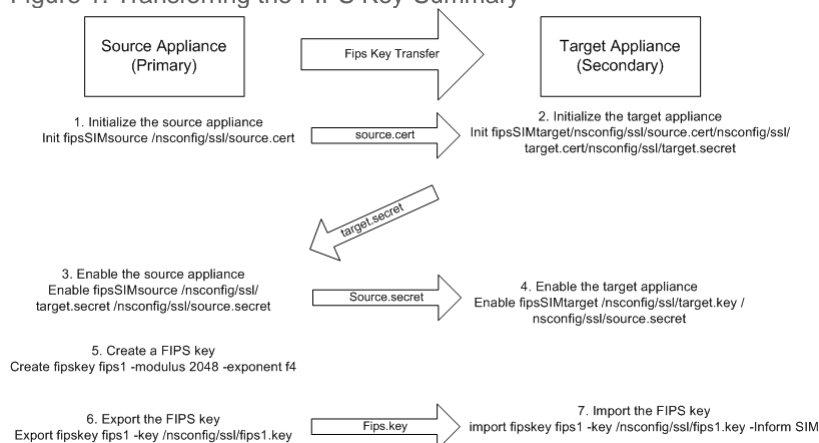
Copy this key into the hard disk of the target appliance.

### On the target appliance

```
import fipskey fips1 -key /nsconfig/ssl/fips1.key
```

The following diagram summarizes the transfer process.

Figure 1. Transferring the FIPS Key-Summary



## Updating the Firmware to Version 2.2 on a FIPS Card

Note: This feature is available from release 10.5, build 55.8007.e and is only available in 10.5.e builds.

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

- TLS protocol versions 1.1 and 1.2 are supported only on SSL virtual servers and front-end SSL services.
- 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 servers, internal, and front end services. To use TLS 1.1/1.2, you must explicitly enable these protocols, on the SSL entities, after the upgrade.

## Prerequisites

1. Download the CN16XX-FW-2.2.tar tarball from the download page on [www.citrix.com](http://www.citrix.com).
2. Extract the contents. For example: `tar -xvf CN16XX-FW-2.2.tar`

After extraction, a folder CN16XX-FW-2.2 is created containing the following 2 files:

- FW 2.2 File: CN16XX-NFBE-FW-2.2-130009
- FW 2.2 Signature File: CN16XX-NFBE-FW-2.2-130009.sign

Note: To verify that the files are extracted correctly, run md5 on both the files and make sure that it matches the following:

- `$ md5 CN16XX-NFBE-FW-2.2-130009`

MD5 (CN16XX-NFBE-FW-2.2-130009) = 0a773c3709c9fd280349c0a38dde445c

- `$ md5 CN16XX-NFBE-FW-2.2-130009.sign`

MD5 (CN16XX-NFBE-FW-2.2-130009.sign) = 131388e39a347490db532da3b12cafa8

## To 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

3. Save the configuration. At the prompt, type:

```
save config
```

4. Perform the update. At the prompt, type:

```
update ssl fips -fipsFW <path to the extracted contents>/CN16XX-NFBE-FW-2.2-130009
```

and press **y** when the following prompt appears:

```
This command will update compatible version of the FIPS firmware. You must save the cu
Done
```

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.

5. Restart the appliance. At the prompt, type:

```
reboot
```

```
Are you sure you want to restart NetScaler (Y/N)? [N]:Y
```

6. Verify that the update is successful. At the prompt, type:

```
show fips
```

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-2.2-130009
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

## To 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 [To update the FIPS firmware to version 2.2 on a standalone NetScaler](#).

Force the secondary node to become primary. At the prompt, type:

```
force failover
```

and press **y** at the confirmation prompt.

2. Log on to the new secondary node (old primary) and perform the update as described in [To update the FIPS firmware to version 2.2 on a standalone NetScaler](#).
3. Force the new secondary node to become primary again. At the prompt, type:

```
force failover
```

and press **y** at the confirmation prompt.

## Resetting a Locked HSM

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.

### To reset a locked HSM by using the command line interface

At the command prompt, type the following commands to reset and re-initialize a locked HSM:

- o reset ssl fips
- o reboot -warm
- o set ssl fips -initHSM Level-2 <new SO password> <old SO password> <user password> [-hsmLabel <string>]
- o save ns config
- o reboot -warm

#### Example

```
reset fips
reboot -warm
set fips -initHSM Level-2 newsopin123 sopin123 userpin123 -hsmLabel NSFIPS
saveconfig
reboot -warm
```

Note: The SO and User passwords are the default passwords.

### To reset a locked HSM by using the configuration utility

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.

## FIPS Approved Algorithms and Ciphers

The FIPS approved algorithms are:

Key-Exchange algorithms

- RSA

Cipher algorithms

- SSL3-DES-CBC3-SHA
- TLS1-AES-256-CBC-SHA
- TLS1-AES-128-CBC-SHA

Note: RC4 (ARC4) is not a FIPS-approved algorithm.

SSL virtual server is marked UP only when default ciphers (FIPS) are configured.

## Support for Thales nShield® HSM

Note: This feature is available from release 10.5, build 52.1115.e.

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 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 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 ADC 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.

This section includes the following details:

- [Architecture Overview](#)
- [Prerequisites](#)
- [Configuring the ADC-Thales Integration](#)
- [Limitations](#)
- [Appendix](#)

## Architecture Overview

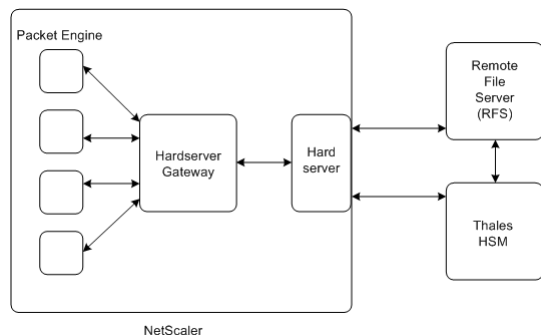
The three entities that are part of a NetScaler-Thales deployment are a Thales nShield Connect module, a remote file server (RFS), and a NetScaler ADC.

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 documentation.

The following figure illustrates the interaction between the different components.

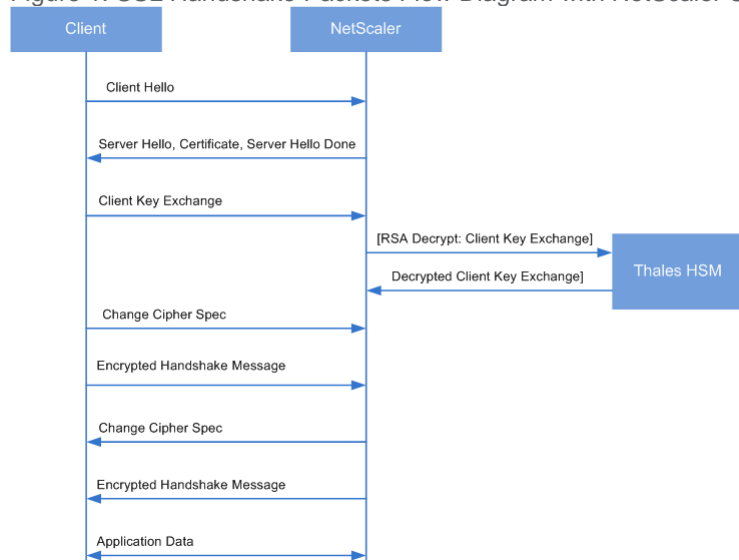


In a typical deployment, the RFS is used to securely store keys generated by the HSM. After the keys are generated, you can securely transfer them to the ADC and then use the NetScaler configuration utility or command line to load the keys to the HSM. A virtual server on the ADC uses Thales to decrypt 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



Note: The communication between the ADC and the HSM uses a Thales proprietary communication protocol, called IMPATH



## Prerequisites

Before you can use a Thales nShield Connect with a NetScaler ADC, make sure that the following prerequisites are met:

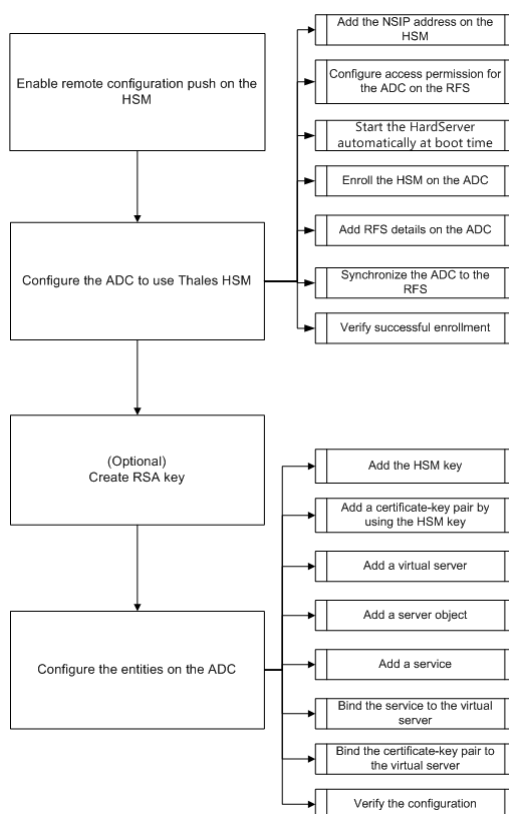
- A Thales nShield Connect device is installed in the network, ready to use, and accessible to the NetScaler ADC. That is, the NetScaler IP (NSIP) address is added as an authorized client on the HSM.
- A usable Security World exists. Security World is a unique key management architecture used by the 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](http://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 ADC.

- Licenses are available to support the number of clients that will be connected to the Thales HSM. The ADC and RFS are clients of the HSM.
- A remote file server (RFS) is installed in the network and is accessible to the NetScaler ADC.
- The Thales nShield Connect device, the RFS, and the NetScaler ADC 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.

## Configuring the ADC-Thales Integration

The following flowchart depicts the tasks that you need to perform to use Thales HSM with a NetScaler ADC:



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.
  - Start the Hardserver.
  - 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.
  - Restart the ADC.
3. (Optional) Create an HSM RSA key.
4. Configure the entities on the NetScaler ADC.
  - 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.

## Enabling 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.

### To 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 NetScaler IP (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>
```

#### Example

```
root@ns# /opt/nfast/bin/anonkneti 10.217.2.112
BD17-C807-58D9 5e30a698f7bab3b2068ca90a9488dc4e6c78d822
```

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
```

In the `hs_clients` section, add the following entries:

```
Amount of data in bytes to encrypt with a session key before session key# renegotiation, o
datalimit=INT
addr=10.217.2.43
clientperm=unpriv
keyhash=00
esn=
timelimit=86400
datalimit=8388608

```

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
```

#### Example

```
/opt/nfast/bin/cfg-pushnethsm --address=10.217.2.112 --force
/opt/nfast/kmdata/hsm-BD17-C807-58D9/config/config
```

### 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>
```

#### 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

Verify that the ADC can reach both the RFS and Thales HSM by using port 9004.

## Start the Hardserver

Change directory to /var/opt/nfast/scripts/startup.

To start the Hardserver, run the following command on the ADC:

```
./hardserver &
```

## 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>)
```

### Example

```
root@ns# ./nethsmenroll --force 10.217.2.112 $(anonkneti 10.217.2.112)
OK configuring hardserver's nethsm imports
```

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.

```
â€|
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=
```

Change directory to /var/opt/nfast/bin and run the following command on the ADC:

```
touch "thales_hsm_is_enrolled"
```

Note: To remove an HSM that is enrolled on the ADC, type: ./nethsmenroll â€"remove <NETHSM-IP>

## 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>
```

### 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.
```

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.

```
â€|â€|
remote_ip=10.217.2.6
remote_port=9004
use_kneti=no
local_esn=
```

Note: To remove an RFS that is enrolled on the ADC, type: `./rfs_sync -remove`

#### 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.
```

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=
```

## 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
```

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# ./chkserver root@ns# ./nfmkminfo root@ns# ./sigtest
```

See [Appendix](#) for an example.

## Restart the ADC

You must restart the ADC for the configuration to take effect. If you have made changes to the configuration, save the configuration before you restart the ADC.

### To restart the ADC by using the command line

At the command prompt, run the following command:

```
reboot
```

### To restart the ADC by using the configuration utility

1. In the configuration utility, 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.

## Create an HSM RSA Key

Updated: 2014-12-08

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
nvrn: 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
nvrn 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
```

### 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\_2ed5428aaeae1e159bdbd63f25292c7113ec2c78)

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,
2ed5428aaeae1e159bdbd63f25292c7113ec2c78)
[default c6410ca00af7e394157518cb53b2db46ff18ce29]
> 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
ident: Key identifier? [] > examplrsa2048key
plainname: Key name? [] > examplrsa2048key
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 2ed5428aaeae1e159bdbd63f25292c7113ec2c78
ident Key identifier exemplersa2048key
plainname Key name exemplersa2048key
Key successfully retargetted.
Path to key: /opt/nfast/kmdata/local/key_simple_exemplersa2048key

```

Important! When prompted for the source key identifier, enter **2ed5428aaeae1e159bdbd63f25292c**

## Result

A key with the prefix `key_simple` (for example `key_simple_exemplersa2048key`) is created.

Note: `exemplersa2048key` 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

Updated: 2014-12-08

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.

#### To enable a feature by using the command line

At the command prompt, run the following commands:

- o `enable feature lb`
- o `enable feature ssl`

#### To enable a feature by using the configuration utility

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](#).

#### To add a SNIP address and verify the configuration by using the command line

At the command prompt, run the following commands:

- o `add ns ip <IPAddress> <netmask> -type SNIP`
- o `show ns ip`

#### 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 |
|----|----------------|----------------|--------------|--------|---------|---------|---------|
|    | -----          | -----          | ----         | ----   | ---     | ----    | -----   |
| 1) | 192.168.17.251 | 0              | NetScaler IP | Active | Enabled | Enabled | NA      |

|      |                |   |      |        |         |         |         |
|------|----------------|---|------|--------|---------|---------|---------|
| 2)   | 192.168.17.252 | 0 | VIP  | Active | Enabled | Enabled | Enabled |
| 3)   | 192.168.17.253 | 0 | SNIP | Active | Enabled | Enabled | NA      |
| Done |                |   |      |        |         |         |         |

### To add a SNIP address and verify the configuration by using the configuration utility

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.

### To add an HSM key by using the command line

At the shell prompt, run the following command:

```
add ssl hsmKey <hsmKeyName> -key <string>
```

#### Example

```
> add ssl hsmKey examplersa2048key -key "key key_simple_examplersa2048key"
Done
```

### To add an HSM key by using the configuration utility

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 [Adding or Updating a Certificate-Key Pair](#).

### To add a certificate-key pair by using the command line

At the command prompt, run the following command:

```
add ssl certKey <certkeyName> -cert <string> -hsmKey <string>
```

#### Example

```
> add ssl certKey key22 -cert example_selfcert -hsmKey examplersa2048key
Done
```

### To add a certificate-key pair by using the configuration utility

Navigate to Traffic Management > SSL > Certificates, and add a certificate-key pair.

### Add a virtual server

For information about a virtual server, see [Configuring an SSL-Based Virtual Server](#).

### To configure an SSL-based virtual server by using the command line

At the command prompt, run the following command:

```
add lb vserver <name> <serviceType> <IPAddress> <port>
```

#### Example

```
> add lb vserver v1 SSL 192.168.17.252 443
```

### To configure an SSL-based virtual server by using the configuration utility



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
```

### To add a server object by using the command line

At the command prompt, run the following command:

```
add server <name> <IPAddress>
```

### Example

```
> add server s1 192.168.17.246
```

### To add a server object by using the configuration utility

Navigate to Traffic Management > Load Balancing > Servers, and add a server.

## Add a service

For more information, see [Configuring Services](#).

### To configure a service by using the command line

At the command prompt, run the following command:

```
add service <name> <serverName> <serviceType> <port>
```

### Example

```
> add service sr1 s1 HTTP 80
```

### To configure a service by using the configuration utility

Navigate to Traffic Management > Load Balancing > Services, and create a service.

## Bind the service to the virtual server

For more information, see [Binding Services to an SSL-Based Virtual Server](#).

### To bind a service to a virtual server by using the command line

At the command prompt, run the following command:

```
bind lb vserver <name> <serviceName>
```

### Example

```
> bind lb vserver v1 sr1
```

### To bind a service to 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 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 [Binding the Certificate-Key Pair to the SSL-Based Virtual Server](#).

### To bind a certificate-key pair to a virtual server by using the command line

At the command prompt, run the following command:

```
bind ssl vserver <vServerName> -certkeyName <string>
```

### Example

```
> bind ssl vserver v1 -certkeyName key22
Warning: Current certificate replaces the previous binding
```

### To bind a certificate-key pair to a virtual server by using the configuration utility

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 command line

At the command prompt, run the following commands:

- o show lb vserver <name>
- o show ssl vserver <vServerName>

#### 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
RHlstate: 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) srl (192.168.17.246: 80) - HTTP State: UP Weight: 1
Done
>

> 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: DISA
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

**To view the configuration by using the configuration utility**

Navigate to Traffic Management > Load Balancing > Virtual Servers, and double-click an SSL virtual server to open it and view the configuration.

## Limitations

- TLS protocol versions 1.1 and 1.2 are not supported with the Thales HSM integration.  
Note: 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.
- ECDHE/DHE ciphers and 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 NetScaler configuration utility to enroll the ADC as a client of the HSM or check the status of the HSM from the configuration utility.
- SSL renegotiation is not 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.

## Appendix

### Example

Note: In the following example, the commands are in boldface type.

```

root@ns# ./chkserver
nCIPHER server running
root@ns# ./nfkminfo
World
generation 2
state 0x17a70000 Initialised Usable Recovery PINRecovery !ExistingClient RTC NVRAM FT
n_modules 1
hknso cbec8c0c56c6b5e76b73147ef02d34a661eaa044
hkm bbb8d4839da5782be4d092735a7535538834dc91 (type Rijndael)
hkmwk 1d572201be533ebc89f30fdd8f3fac6ca3395bf0
hkrc 01f21ecf43933ffdd45e74c3883525176c5c439c
hkra ac8ec5ee6bce00991bd97adce2091d9739b9b452
hkmc cf1b509abaad91995ed202d8f36613fc99433155
hkp c20910b2ed1ca62d6a2b0db67052a05f7bbfeb43
hkrtc bd811020a7c2f8df435a481c3767a89c2e13bc4f
hknv 278b8012e48910d518a9ee91cff57233fb0c9093
hkdsee 12230b0e31e3cec66324c0815f782cfb9249edd5
hkfto 89dd6250b3d6149bcd15606f4553085e2fd6271a
hkmmnull 0100
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 70289a6edba0ddc7e3f6d6f5a49edc963e822f2
```

```
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_RSAPKCS1 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

## Troubleshooting

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

Updated: 2013-07-22

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 above resources, you can use the Wireshark application customized for the NetScaler trace files to expedite troubleshooting.

### Troubleshooting SSL Issues

Updated: 2013-07-22

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.

