asterisk indicates a required parameter. For a term in parentheses, see the corresponding argument in the table above.)

- **Name*** (`name`; Note: Cannot be changed for a previously configured virtual server)
- **Protocol*** (`type`)
- **IP address*** (`IP`)
- **Port** (`port`)

4. In the **Advanced** tab, expand **Listen Policy**, and then type or select values for the following parameters. (An asterisk indicates a required parameter. For a term in parentheses, see the corresponding argument in the table above.)
   - **Listen Priority*** (`priority`)
   - **Listen Policy Rule*** (`rule`)
5. Click **Create** or **OK**, depending on whether you are creating a new virtual server or modifying an existing virtual server.
6. Click **Close**.

   The virtual server that you created now appears in the **Virtual Servers** page.
7. To remove a virtual server, in the **Virtual Servers** page select the virtual server, and then click **Remove**.

After you have created this virtual server, you bind it to one or more services as described in “Configuring a Basic Setup,” on page 30.

Configuring Persistent Connections Between Clients and Servers

The NetScaler initially selects a server by using a load balancing method. With persistence configured, enabling the NetScaler to send any subsequent client requests to the selected server, the server can access state information for that client.

If persistence is configured, it overrides the load balancing methods once the server has been selected. If the configured persistence applies to a service that is down, the NetScaler uses the load balancing methods to select a new service, and the new service becomes persistent for subsequent requests from the client. If the state selected service is out of service, it continues to serve the outstanding requests, but it does not allow new requests or connections. After the shutdown period elapses, no new requests or connections are directed to the service and the existing connections are closed.

You can configure different types of persistence on the NetScaler.

Persistence types

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

Configuring Persistence Types

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

Relationship of Persistence Type to Virtual Server Type

Persistence type	HTTP	HTTPS	TCP	User Datagram Protocol (UDP)/IP	SSL_Bridge
Source IP	YES	YES	YES	YES	YES
CookieInsert	YES	YES	NO	NO	NO
SSL Session ID	NO	YES	NO	NO	YES
URL passive	YES	YES	NO	NO	NO
Custom Server ID	YES	YES	NO	NO	NO
Rule	YES	YES	NO	NO	NO
SRCIPDESTIP	NA	NA	YES	YES	NA
DESTIP	NA	NA	YES	YES	NA

To configure persistence on a virtual server by using the NetScaler command line

At the NetScaler command prompt, type:

```
set lb vserver <name> -PersistenceType <type>
```

Example

```
set lb vserver Vserver-LB-1 -persistenceType SOURCEIP
```

Persistence Parameters

Parameters	Specifies
Persistence Type (`persistenceType`)	Persistence type for the virtual server. The valid options for this parameter are: SOURCEIP, COOKIEINSERT, SSLSESSION, RULE, URLPASSIVE, CUSTOMSERVERID, DESTIP, SRCIPDESTIP, CALLID, and NONE (default)
Persistence Mask (`persistMask`)	Persistence Mask is used to specify if the persistence is IP-based. The default value is 255.255.255.255. If you set 0 using this parameter the complete IP address is used for persistence.
Time-out (`timeout`)	The time period for which persistence is in effect for a specific client. The default value is 2 minutes, and the maximum value that can be configured is 1440 minutes.

To configure persistence on a virtual server by using the configuration utility

1. In the navigation pane, expand **Load Balancing**, and then click **Virtual Servers**.
2. In the details pane, select the virtual server for which you want to configure persistence (for example, **Vserver-LB-1**), and then click **Open**.
3. In the **Configure Virtual Server (Load Balancing)** dialog box, on the **Method and Persistence** tab, in the **Persistence** list, select the persistence type you want to use (for example, **SOURCEIP**).
4. In the **Time-out** and **Netmask** text boxes type the time-out and netmask values (for example, **2** and **255.255.255.255**).
5. Click **OK**.

Note: After configuring persistence for a virtual server, you can view the persistence type by viewing the virtual server from the configuration utility or using the `show lb vserver` command. You can also use the `show ns persistencesession` command to view persistence sessions.

Configuring Persistence Based on Source IP Addresses

The NetScaler selects a service based on the load balancing method and uses the source IP address of the selected service to send the subsequent requests. The NetScaler creates a session between the client and the servers using the IP address. Using persistence based on source IP address overloads the servers because the connections are routed through the single gateway. In the multiple proxy environments, the client requests arrive at a Web site with different IP addresses. This restriction is known as Mega Proxy problem. You can use CookieInsert persistence to solve this problem.

You can set a time-out value for this type of persistence that specifies the inactivity period for the session. When the time-out value expires, the session is discarded, and a new server is selected based on the configured load balancing method.

To configure persistence based on Source IP Addresses, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **SOURCEIP**.

Configuring Persistence Based on HTTP Cookies

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

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

where

- *NSC_XXXX* is the virtual server ID that is derived from the virtual server name.
- *ServiceIP* is a representation of the IP address of the service.
- *ServicePort* is a representation of the port of the service.

ServiceIP and *ServicePort* are encrypted and inserted. When the NetScaler receives a cookie, it decrypts the cookie.

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

By default, the NetScaler sends an HTTP cookie with version 0, in compliance with the Netscape specification. The NetScaler can also send HTTP cookies with version 1, in compliance with RFC 2109.

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

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

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

When you set the time-out value to 0, the NetScaler does not specify the expiration time regardless of the HTTP cookie version used. The expiration time depends on the client software and such cookies are not valid when the software is shut down. This persistence type does not consume any NetScaler resources. Therefore, it can accommodate an unlimited number of persistent clients.

To configure persistence based on HTTP Cookie, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **COOKIEINSERT**.

Configuring Persistence Based on SSL Session IDs

The NetScaler creates a session-based persistence on the arriving SSL Session ID that is part of the SSL handshake process. The requests with the same SSL session ID are directed to the initially selected service. This persistence is used for SSL bridge type of services, and the NetScaler does not encrypt or decrypt data when it forwards the requests to the services. The NetScaler must maintain the data structures to keep track of the sessions. Thus, the persistence type consumes NetScaler resources. Also, if the SSL sessions renegotiate the session IDs during the transactions, the persistence based on SSL session ID does not function correctly.

The time-out value for this type of persistence is as described in the section "Configuring Persistence Based on Source IP Addresses," on page 102. To configure persistence based on SSL session IDs, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **SSLSESSION**.

Configuring Persistence Based on User-Defined Rules

The NetScaler maintains persistence based on the contents of the matched rule. This persistence type requires configuration of a classic expression that evaluates the request payload or a policy infrastructure expression that evaluates requests or responses. For example, you can configure persistence based on application session information in a response cookie or header. You can configure persistence rules using classic or advanced expressions. For details on these types of expressions and their parameters, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

If a request matches the criteria in the expression, a persistence session is created and subsequent client requests are directed to the previously selected server. To configure rule-based persistence, use the parameters described in the following table.

Parameters for Rule-Based Persistence

Parameter	Specifies
Rule (`rule`)	Value used to set the RULE persistence type. The value can be an existing rule name, or it can be a classic or advanced expression. The default value is none. The maximum length is 14999. The rule evaluates either requests that are directed to the load balanced servers or responses from the servers.
Response Rule (`resRule`)	Value used to set the RULE persistence type. The response rule evaluates responses from the load balanced servers. The value is an advanced expression. The default value is none. The maximum length is 14999. Response rule is not supported for 9.2 nCore.

Example: Classic Expression for a Request Payload

The expression, “httpheader user agent contains MyBrowser” directs any client requests containing, “user agent: MyBrowserV1.1” to the initially selected server.

Example: Advanced Expression for a Request Header

The expression, “HTTP.REQ.HEADER (“user agent”).CONTAINS (“MyBrowser”) directs client requests containing, “user-agent: MyBrowserV1.1” to the initially selected server.

The time-out value for this type of persistence is as described in the section “Configuring Persistence Based on Source IP Addresses,” on page 102.

Example: Advanced Expression for a Response Cookie

The expression, “HTTP.RES.HEADER("SET-COOKIE").VALUE(0).TYPECAST_NVLIST_T('=',';').VALUE("server")” examines server responses containing “server” cookies to the initially selected server.

Note: Before you configure a persistence type at the NetScaler command line, you must perform the steps described in the section "Configuring Persistence Types," on page 100.

To configure persistence based on user-defined rules by using the command line

At the NetScaler command line, type:

```
set lb vserver <vserverName> [-rule expression][-resRule
expression]
```

Examples

```
set lb vserver vsvr_name –rule
http.req.header("cookie").value(0).typecast_nvlist_t('=',';').value
("server")
```

```
set lb vserver vsvr_name –resrule
http.res.header("set-cookie").value(0).typecast_nvlist_t('=',';').v
alue("server")
```

To configure persistence based on user-defined rules by using the configuration utility

1. Perform the command-line version steps described in the section "Configuring Persistence Types," on page 100.
2. In the **Persistence** list, select **RULE**.
3. To configure a rule that analyzes requests, click the **Configure** button next to the **Rule** field. To configure a rule that analyzes responses from the server, click the **Configure** button next to the **Response Rule** field.
4. Select **Classic Syntax** or **Advanced Syntax**, and configure the rule. For more information, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

Configuring Persistence based on Server IDs in URLs

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

URL passive persistence requires configuring either a payload expression or a policy infrastructure expression specifying the location of the server ID in the client requests. You can configure either classic or advanced expressions for this type of persistence. However, to support an IPv6 address for URL passive persistence, you must use an advanced expression. For more information about classic and advanced policy expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

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

Example: Payload Expression

The expression, `URLQUERY contains sid=` configures the NetScaler to extract the server ID from the URL query of a client request, after matching token `sid=`. Thus, a request with the URL http://www.citrix.com/ index.asp?&sid=c0a864100050 is directed to the server with the IP address 10.102.29.10 and port 80.

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

To configure persistence based on server IDs in a URL, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **URLPASSIVE**.

Configuring Persistence Based on Server IDs in Client Requests

The NetScaler maintains persistence based on the server ID that you provide. The persistence type requires you to provide the server ID and embeds the server ID in the response. The NetScaler extracts the server ID from subsequent requests and uses it to choose a server. The server ID value must be in the range 0 through 65535. You must configure the same number in the NetScaler for the corresponding service. This persistence type is called Custom Server ID persistence.

The Custom Server ID persistence type requires a payload expression, or an advanced expression to be configured that specifies the location of the server ID in the client requests. For more information about expressions, see the *Citrix NetScaler Policy Configuration and Reference Guide*.

Note the following:

- If a server ID cannot be extracted from the client requests, server selection is performed based on the load balancing method.
- Avoid using the same server ID for multiple services.

Example: Payload Expression

Two servers, **Server-1** and **Server-2**, are configured, where **Server-1** has the **Server-ID=5** and **Server-2** has the **Server-ID=6**. The services for these servers are bound to the virtual server **Vserver-LB-1**.

The expression "URLQUERY contains sid=" configures the NetScaler to extract the Server-ID from the URL query of the client requests, after matching token "sid=". Thus, a request with the URL `http://www.citrix.com/index.asp?&sid=6` that arrives on **Vserver-LB-1** is directed to the server **Server-2**.

The time-out value does not affect this persistence type. Persistence is maintained for as long as a server ID can be extracted from the client requests. This persistence type does not consume any NetScaler resources, so it can accommodate an unlimited number of persistent clients.

To configure persistence based on server IDs in client requests, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **CUSTOMSERVERID**.

Configuring Persistence Based on Destination IP Addresses

This persistence type is used with link load balancing. With destination IP addresses (DESTIP) configured, the NetScaler chooses a service based on the configured load balancing method. It then directs all subsequent packets to the selected service.

The time-out value for this type of persistence is as described in the section "Configuring Persistence Based on Source IP Addresses," on page 102. To configure persistence based on destination IP addresses, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **DESTIP**.

Configuring Persistence Based on Source and Destination IP Addresses

With source and destination IP addresses (SRCIPDESTIP) configured, the NetScaler chooses a service based on the configured load balancing method and directs subsequent packets with the same source and destination IP addresses to the same service.

The time-out value for this type of persistence is as described in the section "Configuring Persistence Based on Source IP Addresses," on page 102. To configure persistence based on source and destination IP addresses, perform the steps described in the section "Configuring Persistence Types," on page 100. In the **Persistence** list, select **SRCIPDESTIP**.

Configuring Persistence Based on RTSP Session IDs

The NetScaler uses Real-Time Streaming Protocol (RTSP) session ID persistence for Real-Time Streaming Protocol (RTSP) virtual servers, and you cannot change this setting on RTSP virtual servers. The NetScaler selects an RTSP service based on the load balancing method and uses the RTSP session ID to send the subsequent requests. RTSP is stateful, and when the client issues a SETUP command to the server, the server negotiates RTSP session IDs in the SETUP response message. The NetScaler creates a session between the client and the servers by using the RTSP session ID. The RTSP requests and responses must have the session ID header to identify the session.

Sometimes multiple servers can have the same session ID, and, therefore, unique sessions cannot be created between the client and the server. In such cases, you can configure the NetScaler to append the server IP address and port to the session ID so that the session ID is unique. The following table describes the parameter that the NetScaler can use to append the server IP address and port.

Parameters for RTSP Session ID-Based Persistence

Parameter	Specifies
Session ID Mapping (`session`)	Map RTSP session ID by appending the IP address and port of the server to the session ID. The parameter is enabled on a service and the non-persistent data connections are not routed through the NetScaler. When the setting is enabled, the NetScaler rejects any request that does not contain the prefix. Possible values: ON and OFF. Default: OFF.

Note: If the client sends multiple SETUP requests on one TCP connection, the NetScaler sends the SETUP requests to the same server because the NetScaler makes the load balancing decision for every TCP connection. In this case, the NetScaler does not forward the SETUP requests to different servers based on the session ID.