

BIG-IP® Local Traffic Manager™: Implementations

Version 11.5.1



Table of Contents

Configuring a Simple Intranet.....	15
Overview: A simple intranet configuration.....	15
Task summary.....	15
Creating a pool.....	16
Creating a virtual server.....	16
Configuring ISP Load Balancing.....	17
Overview: ISP load balancing.....	17
Illustration of ISP load balancing.....	17
Task summary for ISP load balancing.....	17
Creating a load balancing pool.....	18
Creating a virtual server for inbound content server traffic.....	18
Creating a virtual server for outbound traffic for routers.....	19
Creating self IP addresses an external VLAN.....	19
Enabling SNAT automap for internal and external VLANs.....	20
Routing Based on XML Content.....	21
Overview: XML content-based routing.....	21
Task summary.....	21
Creating a custom XML profile.....	22
Writing XPath queries.....	23
Creating a pool to manage HTTP traffic.....	23
Creating an iRule.....	24
Viewing statistics about XML content-based routing.....	25
Configuring nPath Routing.....	27
Overview: Layer 2 nPath routing.....	27
About Layer 2 nPath routing configuration.....	28
Guidelines for UDP timeouts.....	28
Guidelines for TCP timeouts.....	28
Task summary.....	29
Creating a custom Fast L4 profile.....	29
Creating a server pool for nPath routing.....	29
Creating a virtual server for Layer 2 nPath routing.....	30
Configuring the virtual address on the server loopback interface.....	30
Setting the route for inbound traffic.....	30
Configuring Layer 3 nPath Routing.....	31

Overview: Layer 3 nPath routing.....	31
Configuring Layer 3 nPath routing using tmsh.....	31
Configuring a Layer 3 nPath monitor using tmsh.....	32
Layer 3 nPath routing example.....	33
Creating a Basic Web Site and E-commerce Configuration.....	35
Overview: Basic web site and eCommerce configuration.....	35
Illustration of basic web site and eCommerce configuration.....	35
Task summary.....	35
Creating a pool to process HTTP traffic.....	36
Creating a pool to manage HTTPS traffic.....	36
Creating a virtual server to manage HTTP traffic.....	37
Creating a virtual server to manage HTTPS traffic.....	38
Installing a BIG-IP System Without Changing the IP Network.....	39
Overview: Installing a BIG-IP system without changing the IP network.....	39
Task summary.....	40
Removing the self IP addresses from the default VLANs.....	40
Creating a VLAN group.....	40
Creating a self IP for a VLAN group.....	41
Creating a pool of web servers.....	41
Creating a virtual server.....	41
Enabling IP Address Intelligence.....	43
Overview: Enabling IP address intelligence.....	43
Enabling IP address intelligence.....	43
Creating an iRule to log IP address intelligence information.....	44
Creating an iRule to reject requests with questionable IP addresses.....	45
Checking the reputation of an IP address.....	46
Checking the status of the IP intelligence database.....	46
IP address intelligence categories.....	46
Managing Client-side HTTPS Traffic Using a Self-signed Certificate.....	49
Overview: Managing client-side HTTPS traffic using a self-signed certificate.....	49
Task summary.....	49
Creating a self-signed SSL certificate.....	49
Creating a custom HTTP profile.....	50
Creating a custom Client SSL profile.....	50
Creating a pool to process HTTP traffic.....	51
Creating a virtual server for client-side HTTPS traffic.....	51
Implementation result.....	52
Managing Client and Server HTTPS Traffic using a Self-signed Certificate.....	53

Overview: Managing client and server HTTPS traffic using a self-signed certificate.....	53
Task summary.....	53
Creating a self-signed SSL certificate.....	53
Creating a custom HTTP profile.....	54
Creating a custom Client SSL profile.....	54
Creating a custom Server SSL profile.....	55
Creating a pool to manage HTTPS traffic.....	55
Creating a virtual server for client-side and server-side HTTPS traffic.....	56
Implementation results.....	57
 Securing HTTP Traffic Using a Self-signed Certificate with an Elliptic Curve DSA	
Key.....	59
Overview: Managing client-side HTTP traffic using a self-signed, ECC-based certificate.....	59
Task summary.....	59
Creating a self-signed SSL certificate.....	59
Creating a custom HTTP profile.....	60
Creating a custom Client SSL profile.....	60
Creating a pool to process HTTP traffic.....	61
Creating a virtual server for client-side HTTPS traffic.....	61
Implementation results.....	62
 Managing Client-side HTTPS Traffic using a CA-signed Certificate.....	63
Overview: Managing client-side HTTPS traffic using a CA-signed certificate.....	63
Task summary.....	63
Requesting a certificate from a certificate authority.....	63
Creating a custom HTTP profile.....	64
Creating a custom Client SSL profile.....	64
Creating a pool to process HTTP traffic.....	65
Creating a virtual server for client-side HTTPS traffic.....	66
Implementation results.....	66
 Securing HTTP Traffic using a CA-signed Certificate with an Elliptic Curve DSA Key.....	67
Overview: Managing client-side HTTP traffic using a CA-signed, ECC-based certificate.....	67
Task summary.....	67
Requesting a signed certificate that includes an ECDSA key.....	67
Creating a custom HTTP profile.....	68
Creating a custom Client SSL profile.....	69
Creating a pool to process HTTP traffic.....	69
Creating a virtual server for client-side HTTPS traffic.....	70
Implementation results.....	70

Configuring Content Adaptation for HTTP Requests.....	71
Overview: Configuring HTTP Request Adaptation.....	71
Task summary.....	72
Creating a custom client-side ICAP profile.....	72
Creating a pool of ICAP servers.....	73
Creating an internal virtual server for forwarding requests to an ICAP server.....	73
Creating a custom Request Adapt profile.....	74
Creating a custom HTTP profile.....	75
Creating a pool to process HTTP traffic.....	75
Creating an HTTP virtual server for enabling request adaptation.....	76
Implementation result.....	76
 Configuring Content Adaptation for HTTP Requests and Responses.....	 77
Overview: Configuring HTTP Request and Response Adaptation	77
Task summary.....	78
Creating a custom client-side ICAP profile.....	78
Creating a custom server-side ICAP profile.....	79
Creating a pool of ICAP servers.....	80
Creating an internal virtual server for forwarding requests to an ICAP server.....	80
Creating an internal virtual server for forwarding responses to an ICAP server.....	81
Creating a custom Request Adapt profile.....	81
Creating a custom Response Adapt profile.....	82
Creating a custom HTTP profile.....	83
Creating a pool to process HTTP traffic.....	83
Creating an HTTP virtual server for enabling request and response adaptation.....	84
Implementation result.....	84
 Implementing SSL Forward Proxy on a Single BIG-IP System.....	 87
Overview: SSL forward proxy client and server authentication.....	87
Task summary.....	87
Creating a custom Client SSL forward proxy profile.....	88
Creating a custom Server SSL forward proxy profile.....	89
Creating a load balancing pool.....	89
Creating a virtual server for client-side and server-side SSL traffic.....	90
Implementation result.....	91
 Implementing Proxy SSL on a Single BIG-IP System.....	 93
Overview: Direct client-server authentication with application optimization.....	93
Task summary.....	93
Creating a custom Server SSL profile.....	94

Creating a custom Client SSL profile.....	94
Creating a load balancing pool.....	95
Creating a virtual server for client-side and server-side SSL traffic.....	95
Implementation result.....	96
Configuring HTTP Load Balancing with Source Address Affinity Persistence.....	97
Overview: HTTP load balancing with source affinity persistence.....	97
Task summary.....	97
Creating a pool to process HTTP traffic.....	97
Creating a virtual server for HTTP traffic.....	98
Configuring HTTP Load Balancing with Cookie Persistence.....	99
Overview: HTTP load balancing with cookie persistence.....	99
Task summary.....	99
Creating a custom cookie persistence profile.....	99
Creating a pool to process HTTP traffic.....	100
Creating a virtual server for HTTP traffic.....	100
Compressing HTTP Responses.....	103
Overview: Compressing HTTP responses.....	103
Task summary.....	103
Creating a customized HTTP compression profile.....	103
Creating a virtual server for HTTP compression.....	104
Managing HTTP Traffic with the SPDY Profile.....	105
Overview: Managing HTTP traffic with the SPDY profile.....	105
Task summary for managing HTTP and SPDY traffic.....	106
Creating a pool to process HTTP traffic.....	106
Creating an iRule for SPDY requests.....	106
Creating a virtual server to manage HTTP traffic.....	107
Creating a SPDY profile.....	107
Creating a virtual server to manage SPDY traffic.....	108
Using Via Headers to Acquire Information About Intermediate Routers.....	111
Overview: Using Via headers.....	111
Task summary for identifying intermediate information with Via headers.....	111
Identifying information about intermediate proxies with Via headers.....	111
Removing Via headers from requests and responses.....	112
Configuring the BIG-IP System as a Reverse Proxy Server.....	113
Overview: URI translation and HTML content modification.....	113
About URI translation.....	113
Rules for matching requests to URI rules.....	114

About URI Rules.....	114
Introduction to HTML content modification.....	115
Task summary.....	115
Creating a Rewrite profile to specify URI rules.....	115
Creating an HTML profile for tag removal.....	116
Creating pools for processing HTTP traffic.....	117
Creating a local traffic policy.....	117
Creating a virtual server.....	119
Implementation results.....	119
Configuring the BIG-IP System as an MS SQL Database Proxy.....	121
Overview: Configuring LTM as a database proxy.....	121
About database authentication.....	122
About database access configuration.....	122
Creating a custom MS SQL monitor.....	122
Creating a pool of database servers.....	123
Configuring database access by user.....	123
Creating a custom OneConnect profile.....	124
Creating a database proxy virtual server.....	124
Viewing MS SQL profile statistics.....	125
Load Balancing Passive Mode FTP Traffic.....	127
Overview: FTP passive mode load balancing.....	127
Task Summary for load balancing passive mode FTP traffic.....	127
Creating a custom FTP monitor.....	127
Creating a pool to manage FTP traffic.....	129
Creating a virtual server for FTP traffic.....	130
Load Balancing Passive Mode FTP Traffic with Data Channel Optimization.....	131
Overview: FTP passive mode load balancing with data channel optimization.....	131
Task Summary for load balancing passive mode FTP traffic.....	131
Creating a custom FTP profile.....	131
Creating a custom FTP monitor.....	132
Creating a pool to manage FTP traffic.....	133
Creating a virtual server for FTP traffic.....	134
Implementation result.....	135
Referencing an External File from within an iRule.....	137
Overview: Referencing an external file from an iRule.....	137
iRule commands for iFiles.....	137
Task summary.....	138
Importing a file to the BIG-IP system.....	138
Creating an iFile.....	138

Writing an iRule that references an iFile.....	139
Implementation result.....	139
Configuring the BIG-IP System as a DHCP Relay Agent.....	141
Overview: Managing IP addresses for DHCP clients.....	141
About the BIG-IP system as a DHCP relay agent.....	141
Task summary.....	142
Creating a pool of DHCP servers.....	142
Creating a DHCP Relay type virtual server.....	143
Implementation result.....	144
Configuring the BIG-IP System for DHCP Renewal.....	145
Overview: Renewing IP addresses for DHCP clients.....	145
About DHCP renewal	145
Task summary.....	146
Creating a DHCP renewal virtual server.....	146
Implementation result.....	146
Configuring a One-IP Network Topology.....	147
Overview: Configuring a one-IP network topology.....	147
Illustration of a one-IP network topology for the BIG-IP system.....	147
Task summary for a one-IP network topology for the BIG-IP system.....	148
Creating a pool for processing HTTP connections with SNATs enabled.....	148
Creating a virtual server for HTTP traffic.....	148
Defining a default route.....	149
Configuring a client SNAT.....	149
Implementing Health and Performance Monitoring.....	151
Overview: Health and performance monitoring.....	151
Task summary.....	151
Creating a custom monitor.....	152
Creating a load balancing pool.....	152
Creating a virtual server.....	153
Preventing TCP Connection Requests From Being Dropped.....	155
Overview: TCP request queuing.....	155
Preventing TCP connection requests from being dropped.....	155
Setting Connection Limits.....	157
Overview: About connection limits.....	157
Limiting connections for a virtual server, pool member, or node.....	157
Implementation results.....	157

Load Balancing to IPv6 Nodes.....	159
Overview: Load balancing to IPv6 nodes.....	159
Task summary.....	159
Creating a load balancing pool.....	159
Creating a virtual server for IPv6 nodes.....	160
Mitigating Denial of Service Attacks.....	161
Overview: Mitigating Denial of Service and other attacks.....	161
Denial of Service attacks and iRules.....	161
iRules for Code Red attacks.....	161
iRules for Nimda attacks.....	162
Common Denial of Service attacks.....	162
Task summary.....	164
Configuring adaptive reaping.....	164
Setting the TCP and UDP connection timers.....	165
Applying a rate class to a virtual server.....	165
Calculating connection limits on the main virtual server.....	165
Setting connection limits on the main virtual server.....	166
Adjusting the SYN Check threshold.....	166
Configuring Remote CRLDP Authentication.....	167
Overview of remote authentication for application traffic.....	167
Task Summary.....	167
Creating a CRLDP configuration object for authenticating application traffic remotely.....	167
Creating a custom CRLDP profile.....	168
Modifying a virtual server for CRLDP authentication.....	168
Configuring Remote LDAP Authentication.....	171
Overview of remote LDAP authentication for application traffic.....	171
Task Summary.....	171
Creating an LDAP configuration object for authenticating application traffic remotely.....	171
Creating a custom LDAP profile.....	172
Modifying a virtual server for LDAP authentication.....	172
Configuring Remote RADIUS Authentication.....	173
Overview of remote authentication for application traffic.....	173
Task summary for RADIUS authentication of application traffic.....	173
Creating a RADIUS server object for authenticating application traffic remotely.....	173

Creating a RADIUS configuration object for authenticating application traffic remotely.....	174
Creating a custom RADIUS profile.....	174
Modifying a virtual server for RADIUS authentication.....	175
Configuring Remote SSL LDAP Authentication.....	177
Overview of remote SSL LDAP authentication for application traffic.....	177
Task Summary.....	177
Creating an LDAP Client Certificate SSL configuration object.....	177
Creating a custom SSL Client Certificate LDAP profile.....	178
Modifying a virtual server for SSL Client Certificate LDAP authorization.....	178
Configuring Remote SSL OCSP Authentication.....	181
Overview of remote authentication for application traffic.....	181
Task Summary.....	181
Creating an SSL OCSP responder object for authenticating application traffic remotely.....	181
Creating an SSL OCSP configuration object for authenticating application traffic remotely.....	182
Creating a custom SSL OCSP profile.....	182
Modifying a virtual server for SSL OCSP authentication.....	183
Configuring Remote TACACS+ Authentication.....	185
Overview of remote authentication for application traffic.....	185
Task Summary.....	185
Creating a TACACS+ configuration object.....	185
Creating a custom TACACS+ profile.....	186
Modifying a virtual server for TACACS+ authentication.....	187
Configuring Kerberos Delegation.....	189
Overview of remote authentication for application traffic.....	189
Task Summary.....	189
Creating a Kerberos Delegation configuration object.....	189
Creating a Kerberos delegation profile object from the command line.....	190
Creating a load balancing pool.....	190
Creating a virtual server with Kerberos delegation and Client SSL profiles.....	191
Load Balancing Diameter Application Requests.....	193
Overview: Diameter load balancing.....	193
Task summary.....	193
Creating a custom Diameter profile.....	193
Creating a custom Diameter monitor.....	193
Creating a pool to manage Diameter traffic.....	194

Creating a virtual server to manage Diameter traffic.....	194
Configuring the BIG-IP System for Electronic Trading.....	197
Overview: Configuring the BIG-IP system for electronic trading.....	197
Task summary.....	197
Creating a data group list for a FIX profile.....	197
Creating a FIX profile for electronic trading	198
Creating a load balancing pool.....	199
Creating a virtual server for secure electronic trading.....	199
Viewing FIX message statistics.....	200
Implementation result.....	200
Implementing Low-Latency Electronic Trading Functionality.....	201
Overview: Configuring the BIG-IP system for low-latency electronic trading.....	201
Task summary.....	201
Implementing low-latency electronic trading functionality.....	201
Creating a custom Fast L4 profile.....	202
Creating a pool	202
Creating a virtual server for low-latency electronic trading.....	202
Implementation result.....	203
Implementing Video Quality of Experience Functionality.....	205
Overview: Video Quality of Experience profile.....	205
Creating an iRule to collect video Quality of Experience scores.....	205
Creating an iRule to collect static information about video files.....	206
Creating a video Quality of Experience profile.....	207
Creating a pool	207
Creating a video Quality of Experience virtual server.....	207
Securing Client-side SMTP Traffic.....	209
Overview: Securing client-side SMTP traffic.....	209
Task summary.....	209
Creating an SMTPS profile.....	210
Creating a Client SSL profile.....	210
Creating a virtual server and load-balancing pool.....	210
Implementation result.....	211
Controlling Responses to ICMP Echo Requests.....	213
About ICMP echo responses on the BIG-IP system.....	213
Task summary.....	213
Configuring ICMP echo responses for a virtual address.....	214
Communicating virtual server status to a virtual address.....	214
Implementation results.....	215

Legal Notices.....	217
Legal notices.....	217
Acknowledgments.....	218

Configuring a Simple Intranet

Overview: A simple intranet configuration

The simple intranet implementation is commonly found in a corporate intranet (see the following illustration). In this implementation, the BIG-IP® system performs load balancing for several different types of connection requests:

- HTTP connections to the company's intranet web site. The BIG-IP system load balances the two web servers that host the corporate intranet web site, `Corporate.main.net`.
- HTTP connections to Internet content. These are handled through a pair of cache servers that are also load balanced by the BIG-IP system.
- Non-HTTP connections to the Internet.

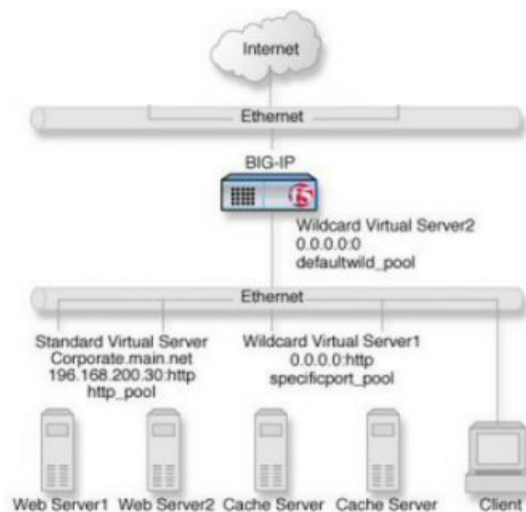


Figure 1: Non-intranet connections

As the illustration shows, the non-intranet connections are handled by wildcard virtual servers; that is, servers with the IP address `0.0.0.0`. The wildcard virtual server that is handling traffic to the cache servers is port specific, specifying port `80` for HTTP requests. As a result, all HTTP requests not matching an IP address on the intranet are directed to the cache server. The wildcard virtual server handling non-HTTP requests is a default wildcard server. A default wildcard virtual server is one that uses only port `0`. This makes it a catch-all match for outgoing traffic that does not match any standard virtual server or any port-specific wildcard virtual server.

Task summary

To create this configuration, you need to complete these tasks.

Task list

Creating a pool

Creating a virtual server

Creating a pool

You can create a pool of servers that you group together to receive and process traffic, to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members. For example, in the illustration, the pool members for **http_pool** are **192.168.100.10:80** and **192.168.100.11:80**.
The pool members for **specificport_pool** are **192.168.100.20:80** and **192.168.100.21:80**.
5. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination** field, verify that the type of virtual server is **Host**, and in the **Address** field, type an IP address for the virtual server.
For example, you can assign the IP address **192.168.200.30:80** to the virtual server that processes HTTP traffic. For load balancing connections to cache servers, you can assign the address **0.0.0.0:80** to the virtual server, making it a wildcard virtual server. To create a forwarding virtual server, you can assign the address **0.0.0.0:0**.
5. In the **Service Port** field, type **80**, or select **HTTP** from the list.
6. In the Configuration area of the screen, locate the **Type** setting and select either **Standard** or **Forwarding (IP)**.
7. From the **HTTP Profile** list, select an HTTP profile.
8. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
9. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Configuring ISP Load Balancing

Overview: ISP load balancing

You might find that as your network grows, or network traffic increases, you require an additional connection to the Internet. You can use this configuration to add an Internet connection to your existing network. The following illustration shows a network configured with two Internet connections.

Illustration of ISP load balancing

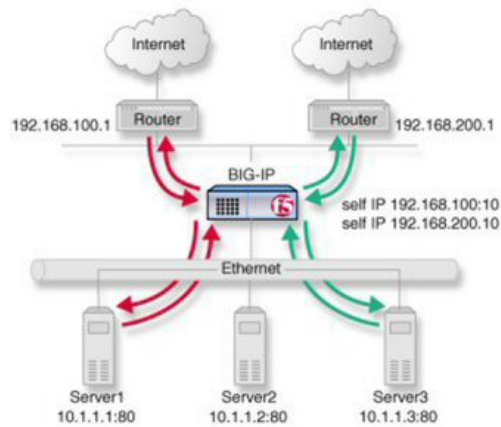


Figure 2: ISP load balancing

Task summary for ISP load balancing

There are number of tasks you must perform to implement load balancing for ISPs.

Task list

Creating a load balancing pool

Creating a virtual server for inbound content server traffic

Creating a virtual server for outbound traffic for routers

Creating self IP addresses an external VLAN

Enabling SNAT automap for internal and external VLANs

Creating a load balancing pool

You can create a load balancing pool, which is a logical set of devices, such as web servers, that you group together to receive and process traffic, to efficiently distribute the load on your resources. Using this procedure, create one pool that load balances the content servers, and one pool to load balance the routers.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

***Tip:** Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Repeat** and create another pool.
9. Click **Finished**.

The load balancing pools appear in the Pools list.

Creating a virtual server for inbound content server traffic

You must create a virtual server to load balance inbound connections. The default pool that you assign as a resource in this procedure is the pool of internal servers.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.

The IP address you type must be available and not in the loopback network.

5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. If the traffic to be load balanced is of a certain type, select the profile type that matches the connection type.
To load balance HTTP traffic, locate the **HTTP Profile** setting and select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. Click **Finished**.

The virtual server is configured to load balance inbound connections to the servers.

Creating a virtual server for outbound traffic for routers

You must create a virtual server to load balance outbound connections. The default pool that you assign as a resource in this procedure is the pool of routers.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
6. Click **Finished**.

The virtual server is configured to load balance outbound connections to the routers.

Creating self IP addresses an external VLAN

You must assign two self IP addresses to the external VLAN.

1. On the Main tab, click **Network > Self IPs**.
2. Click **Create**.
The New Self IP screen opens.
3. In the **IP Address** field, type an IP address.
This IP address should represent the network of the router.
The system accepts IPv4 and IPv6 addresses.
4. In the **Netmask** field, type the full network mask for the specified IP address.
For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
5. Select **External** from the **VLAN** list.
6. Click **Repeat**.
7. In the **IP Address** field, type an IPv4 or IPv6 address.
This IP address should represent the address space of the VLAN that you specify with the **VLAN/Tunnel** setting.
8. Click **Finished**.

The screen refreshes, and displays the new self IP address.

The self IP address is assigned to the external VLAN.

Enabling SNAT automap for internal and external VLANs

You can configure SNAT automapping on the BIG-IP system for internal and external VLANs.

1. On the Main tab, click **Local Traffic > Address Translation**.
The **SNAT List** screen displays a list of existing SNATs.
2. Click **Create**.
3. Name the new SNAT.
4. From the **Translation** list, select **Automap**.
5. For the **VLAN / Tunnel List** setting, in the **Available** field, select **external** and **external**, and using the **Move** button, move the VLANs to the **Selected** field.
6. Click **Finished**.

SNAT automapping on the BIG-IP system is configured for internal and external VLANs.

Routing Based on XML Content

Overview: XML content-based routing

You can use the BIG-IP[®] system to perform XML content-based routing whereby the system routes requests to an appropriate pool, pool member, or virtual server based on specific content in an XML document. For example, if your company transfers information in XML format, you could use this feature to examine the XML content with the intent to route the information to the appropriate department.

You configure content-based routing by creating an XML profile and associating it with a virtual server. In the XML profile, define the matching content to look for in the XML document. Next, specify how to route the traffic to a pool by writing simple iRules[®]. When the system discovers a match, it triggers an iRule event, and then you can configure the system to route traffic to a virtual server, a pool, or a node. You can allow multiple query matches, if needed.

This example shows a simple XML document that the system could use to perform content-based routing. It includes an element called `FinanceObject` used in this implementation.

```
<soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:eai="http://192.168.149.250/eai_enu/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/">
  <soapenv:Header/>
  <soapenv:Body>
    <eai:SiebelEmployeeDelete
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
      <FinanceObject xsi:type="xsd:string">Route to
Financing</FinanceObject>
      <SiebelMessage xsi:type="ns:ListOfEmployeeInterfaceTopElmt"
xmlns:ns="http://www.siebel.com/xml">
        <ListOfEmployeeInterface
xsi:type="ns:ListOfEmployeeInterface">
          <SecretKey>123456789</SecretKey>
          <Employee>John</Employee>
          <Title>CEO</Title>
        </ListOfEmployeeInterface>
      </SiebelMessage>
    </eai:SiebelEmployeeDelete>
  </soapenv:Body>
</soapenv:Envelope>
```

Task summary

You can perform tasks to enable XML content-based routing whereby the system routes requests to an appropriate pool, pool member, or virtual server based on specific content in an XML document.

Task list

Creating a custom XML profile

Writing XPath queries

Creating a pool to manage HTTP traffic

Creating an iRule

Viewing statistics about XML content-based routing

Creating a custom XML profile

To implement content-based routing, you first need to create an XML profile. XML profiles specify the content to look for in XML documents. In the XML profile, you define XPath queries to locate items in an XML document.

1. On the Main tab, click **Local Traffic > Profiles > Services > XML**.
The XML screen opens.
2. Click **Create**.
The New XML screen opens.
3. In the **Name** field, type a unique name for the XML profile, such as `cbr_xml_profile`.
4. In the Settings area, select the **Custom** check box at right.
The settings become available.
5. If you want to reference XML elements with namespaces in XPath queries, from **Namespace Mappings**, select **Specify**.
The screen displays the **Namespace Mappings List** settings.
6. Add namespaces to the list to specify how to map XML namespaces (as defined by the `xmlns` attribute) for the system to use when routing XML traffic to the correct pool, pool member, or virtual server:
 - a) In the **Prefix** field, type the namespace prefix.
 - b) In the **Namespace** field, type the URL that the prefix maps to.
 - c) Click **Add** to add the namespace to the **Namespace Mappings List**.
7. To define the matching criteria in the XML document, from **XPath Queries**, select **Specify**.
The screen displays the **XPath Queries** settings.
8. Add XPath queries to the list to define matching criteria in XML payloads so the system can route the traffic to the correct pool, pool member, or virtual server:
 - a) In the **XPath** field, type an XPath expression.
For example, to look for an element called `FinanceObject`, type `//FinanceObject`.
 - b) Click **Add** to add the XPath expression to the XPath Queries list.
You can define up to three XPath queries.
The expression is added to the list.
9. To allow each query to have multiple matches, select **Multiple Query Matches**.
10. Click **Finished**.
The system creates an XML profile.

You can use the XML profile to route XML traffic. Note that XML profiles do not support use of the Expect header field. This is because the header of a transaction could direct it to one pool, and the payload could invoke an iRule to direct the transaction to a different pool.

Writing XPath queries

You can write up to three XPath queries to define the content that you are looking for in XML documents. When writing XPath queries, you use a subset of the XPath syntax described in the XML Path Language (XPath) standard at <http://www.w3.org/TR/xpath>.

These are the rules for writing XPath queries for XML content-based routing.

1. Express the queries in abbreviated form.
2. Map all prefixes to namespaces.
3. Use only ASCII characters in queries.
4. Write queries to match elements and attributes.
5. Use wildcards as needed for elements and namespaces; for example, `//emp:employee/*`.
6. Do not use predicates in queries.

Syntax for XPath expressions

This table shows the syntax to use for XPath expressions.

Expression	Description
Nodename	Selects all child nodes of the named node.
@Attname	Selects all attribute nodes of the named node.
/	Indicates XPath step.
//	Selects nodes that match the selection no matter where they are in the document.

XPath query examples

This table shows examples of XPath queries.

Query	Description
/a	Selects the root element a.
//b	Selects all b elements wherever they appear in the document.
/a/b:*	Selects any element in a namespace bound to prefix b, which is a child of the root element a.
//a/b:c	Selects elements in the namespace of element c, which is bound to prefix b, and is a child of element a.

Creating a pool to manage HTTP traffic

For implementing content-based routing, you can create one or more pools that contain the servers where you want the system to send the traffic. You write an iRule to route the traffic to the pool.

If you want to specify a default pool to which to send traffic when it does not match the content you are looking for, repeat the procedure to create a second pool. You specify the default pool in the virtual server. Alternatively, you can create a node or a virtual server to route traffic to instead of creating a pool.

1. On the Main tab, click **Local Traffic > Pools**.

The Pool List screen opens.

2. Click **Create**.

The New Pool screen opens.

3. In the **Name** field, type a name for the pool, such as `finance_pool`.

4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:

- Select **Disabled** to disable priority groups. This is the default option.
- Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.

7. Using the **New Members** setting, add each resource that you want to include in the pool:

- a) Type an IP address in the **Address** field.
- b) Type `80` in the **Service Port** field, or select **HTTP** from the list.
- c) (Optional) Type a priority number in the **Priority** field.
- d) Click **Add**.

8. Click **Finished**.

The new pool appears in the Pools list.

Creating an iRule

You create iRules[®] to automate traffic forwarding for XML content-based routing. When a match occurs, an iRule event is triggered, and the iRule directs the individual request to a pool, a node, or virtual server. This implementation targets a pool.

1. On the Main tab, click **Local Traffic > iRules**.

2. Click **Create**.

3. In the **Name** field, type a 1- to 31-character name, such as `XML_CBR_iRule`.

4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.

For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site <http://devcentral.f5.com>.

5. Click **Finished**.

Examples of iRules for XML content-based routing

This example shows an iRule that queries for an element called `FinanceObject` in XML content and if a match is found, an iRule event is triggered. The system populates the values of the Tcl variables (`$XML_count`, `$XML_queries`, and `$XML_values`). Then the system routes traffic to a pool called `finance_pool`.

```
when XML_CONTENT_BASED_ROUTING
{
```

```

for {set i 0} { $i < $XML_count } {incr i} {
    log local0. $XML_queries($i)
    log local0. $XML_values($i)
    if {($XML_queries($i) contains "FinanceObject")} {
        pool finance_pool
    }
}
}

```

This is another example of XML content-based routing. It shows routing by bank name and by price.

```

when XML_CONTENT_BASED_ROUTING
{
    for {set i 0} { $i < $XML_count } {incr i} {
        # routing by BANK_NAME
        if {($XML_queries($i) contains "BANK_NAME")} {
            if {($XML_values($i) contains "InternationalBank")} {
                pool pool1
            } elseif {($XML_values($i) contains "Hapoalim")} {
                pool pool2
            } else {
                pool pool3
            }
        }
        # routing by PRICE
        if {($XML_queries($i) contains "PRICE")} {
            if {($XML_values($i) > 50)} {
                pool pool1
            } else {
                pool pool2
            }
        }
    }
    # end for
}
}

```

Note: The `XML_CONTENT_BASED_ROUTING` event does not trigger when the client's headers contain `"Expect: 100-continue"` regardless of whether the server sends a 100-continue response. In this case, the request is routed to the default pool.

Tcl variables in iRules for XML routing

This table lists and describes the Tcl variables in the sample iRule.

Tcl variable	Description
<code>\$XML_count</code>	Shows the number of matching queries.
<code>\$XML_queries</code>	Contains an array of the matching query names.
<code>\$XML_values</code>	Holds the values of the matching elements.

Viewing statistics about XML content-based routing

You can view statistics about XML content-based routing to make sure that the routing is working.

Note: The system first checks for a match, then checks for malformedness of XML content. So if the system detects a match, it stops checking, and might not detect any subsequent parts of the document that are malformed.

1. On the Main tab, click **Statistics > Module Statistics > Local Traffic**.
The Local Traffic statistics screen opens.
2. From the **Statistics Type** list, select **Profiles Summary**.
3. In the Global Profile Statistics area, for the Profile Type **XML**, click **View** in the Details.
The system displays information about the number of XML documents that were inspected, the number of documents that had zero to three matches, and the number of XML documents that were found to be malformed.

Configuring nPath Routing

Overview: Layer 2 nPath routing

With the Layer 2 nPath routing configuration, you can route outgoing server traffic around the BIG-IP® system directly to an outbound router. This method of traffic management increases outbound throughput because packets do not need to be transmitted to the BIG-IP system for translation and then forwarded to the next hop.

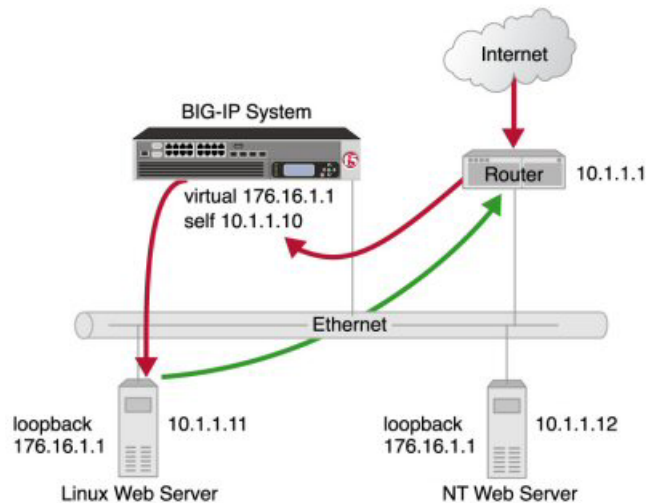


Figure 3: Layer 2 nPath routing

Note: The type of virtual server that processes the incoming traffic must be a transparent, non-translating type of virtual server.

In bypassing the BIG-IP system on the return path, Layer 2 nPath routing departs significantly from a typical load-balancing configuration. In a typical load-balancing configuration, the destination address of the incoming packet is translated from that of the virtual server to that of the server being load balanced to, which then becomes the source address of the returning packet. A default route set to the BIG-IP system then sees to it that packets returning to the originating client return through the BIG-IP system, which translates the source address back to that of the virtual server.

Note: Do not attempt to use nPath routing for Layer 7 traffic. Certain traffic features do not work properly if Layer 7 traffic bypasses the BIG-IP system on the return path.

About Layer 2 nPath routing configuration

The Layer 2 nPath routing configuration differs from the typical BIG-IP® load balancing configuration in the following ways:

- The default route on the content servers must be set to the router's internal address (**10.1.1.1** in the illustration) rather than to the BIG-IP system's floating self IP address (**10.1.1.10**). This causes the return packet to bypass the BIG-IP system.
- If you plan to use an nPath configuration for TCP traffic, you must create a Fast L4 profile with the following custom settings:
 - Enable the **Loose Close** setting. When you enable this setting, the TCP protocol flow expires more quickly, after a TCP FIN packet is seen. (A FIN packet indicates the tearing down of a previous connection.)
 - Set the **TCP Close Timeout** setting to the same value as the profile idle timeout if you expect half closes. If not, you can set this value to 5 seconds.
- Because address translation and port translation have been disabled, when the incoming packet arrives at the pool member it is load balanced to the virtual server address (**176.16.1.1** in the illustration), not to the address of the server. For the server to respond to that address, that address must be configured on the loopback interface of the server and configured for use with the server software.

Guidelines for UDP timeouts

When you configure nPath for UDP traffic, the BIG-IP® system tracks packets sent between the same source and destination address to the same destination port as a connection. This is necessary to ensure the client requests that are part of a session always go to the same server. Therefore, a UDP connection is really a form of persistence, because UDP is a connectionless protocol.

To calculate the timeout for UDP, estimate the maximum amount of time that a server transmits UDP packets before a packet is sent by the client. In some cases, the server might transmit hundreds of packets over several minutes before ending the session or waiting for a client response.

Guidelines for TCP timeouts

When you configure nPath for TCP traffic, the BIG-IP® system recognizes only the client side of the connection. For example, in the TCP three-way handshake, the BIG-IP system sees the SYN from the client to the server, and does not see the SYN acknowledgment from the server to the client, but does see the acknowledgment of the acknowledgment from the client to the server. The timeout for the connection should match the combined TCP retransmission timeout (RTO) of the client and the node as closely as possible to ensure that all connections are successful.

The maximum initial RTO observed on most UNIX and Windows® systems is approximately 25 seconds. Therefore, a timeout of 51 seconds should adequately cover the worst case. When a TCP session is established, an adaptive timeout is used. In most cases, this results in a faster timeout on the client and node. Only in the event that your clients are on slow, lossy networks would you ever require a higher TCP timeout for established connections.

Task summary

There are several tasks you perform to create a Layer 2 nPath routing configuration.

Task list

Creating a custom Fast L4 profile

Creating a server pool for nPath routing

Creating a virtual server for Layer 2 nPath routing

Configuring the virtual address on the server loopback interface

Setting the route for inbound traffic

Creating a custom Fast L4 profile

You can create a custom Fast L4 profile to manage Layer 4 traffic more efficiently.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Protocol** > **Fast L4**.
The Fast L4 screen opens.
2. Click **Create**.
The New Fast L4 profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. Select the **Loose Close** check box.
6. Set the **TCP Close Timeout** setting, according to the type of traffic that the virtual server will process.
7. Click **Finished**.

The custom Fast L4 profile appears in the list of Fast L4 profiles.

Creating a server pool for nPath routing

After you create a custom Fast L4 profile, you need to create a server pool.

1. On the Main tab, click **Local Traffic** > **Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.

d) Click **Add**.

6. Click **Finished**.

Creating a virtual server for Layer 2 nPath routing

After you create a server pool, you need to create a virtual server that references the profile and pool you created.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. From the **Configuration** list, select **Advanced**.
6. From the **Type** list, select **Performance (Layer 4)**.
7. From the **Protocol** list, select one of the following:
 - **UDP**
 - **TCP**
 - *** All Protocols**
8. From the **Protocol Profile (Client)** list, select a predefined or user-defined Fast L4 profile.
9. For the **Address Translation** setting, clear the **Enabled** check box.
10. For the **Port Translation** setting, clear the **Enabled** check box.
11. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
12. Click **Finished**.

Configuring the virtual address on the server loopback interface

You must place the IP address of the virtual server (**176.16.1.1** in the illustration) on the loopback interface of each server. Most UNIX variants have a loopback interface named **lo0**. Consult your server operating system documentation for information about configuring an IP address on the loopback interface. The loopback interface is ideal for the nPath configuration because it does not participate in the ARP protocol.

Setting the route for inbound traffic

For inbound traffic, you must define a route through the BIG-IP[®] system self IP address to the virtual server. In the example, this route is **176.16.1.1**, with the external self IP address **10.1.1.10** as the gateway.

Note: You need to set this route only if the virtual server is on a different subnet than the router.

For information about how to define this route, please refer to the documentation provided with your router.

Configuring Layer 3 nPath Routing

Overview: Layer 3 nPath routing

Using Layer 3 nPath routing, you can load balance traffic over a routed topology in your data center. In this deployment, the server sends its responses directly back to the client, even when the servers, and any intermediate routers, are on different networks. This routing method uses IP encapsulation to create a uni-directional outbound tunnel from the server pool to the server.

You can also override the encapsulation for a specified pool member, and either remove that pool member from any encapsulation or specify a different encapsulation protocol. The available encapsulation protocols are IP and GRE.

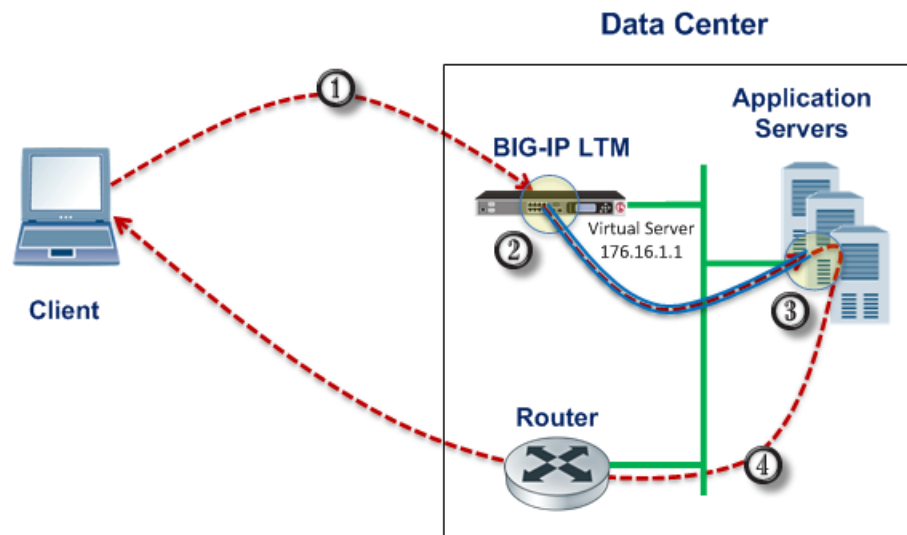


Figure 4: Example of a Layer 3 routing configuration

This illustration shows the path of a packet in a deployment that uses Layer 3 nPath routing through a tunnel.

1. The client sends traffic to a Fast L4 virtual server.
2. The pool encapsulates the packet and sends it through a tunnel to the server.
3. The server removes the encapsulation header and returns the packet to the network.
4. The target application receives the original packet, processes it, and responds directly to the client.

Configuring Layer 3 nPath routing using tmsh

Before performing this procedure, determine the IP address of the loopback interface for each server in the server pool.

Use Layer 3 nPath routing to provide direct server return for traffic in a routed topology in your data center.

1. On the BIG-IP® system, start a console session.
2. Create a server pool with an encapsulation profile.

```
tmsh create ltm pool npath_ipip_pool profiles add  
{ ipip } members add { 10.7.1.7:any 10.7.1.8:any 10.7.1.9:any }
```

This command creates the pool `npath_ipip_pool`, which has three members that specify all services: `10.7.1.7:any`, `10.7.1.8:any`, and `10.7.1.9:any`, and applies IPIP encapsulation to outbound traffic.

3. Create a profile that disables hardware acceleration.

```
tmsh create ltm profile fastl4 fastl4_npath pva-acceleration none
```

This command disables the Packet Velocity® ASIC acceleration mode in the new Fast L4 profile named `fastl4_npath`.

4. Create a virtual server that has address translation disabled, and includes the pool with the encapsulation profile.

```
tmsh create ltm virtual npath_udp destination 176.16.1.1:any  
pool npath_ipip_pool profiles add { fastl4_npath } translate-address  
disabled ip-protocol udp
```

This command creates a virtual server named `npath_udp` that intercepts all UDP traffic, does not use address translation, and does not use hardware acceleration. The destination address `176.16.1.1` matches the IP address of the loopback interface on each server.

These implementation steps configure only the BIG-IP device in a deployment example. To configure other devices in your network for L3 nPath routing, consult the device manufacturer's documentation for setting up direct server return (DSR) for each device.

Configuring a Layer 3 nPath monitor using tmsh

Before you begin this task, configure a server pool with an encapsulation profile, such as `npath_ipip_pool`.

You can create a custom monitor to provide server health checks of encapsulated tunnel traffic. Setting a variable in the `db` component causes the monitor traffic to be encapsulated.

1. Start at the Traffic Management Shell (tmsh).
2. Create a transparent health monitor with the destination IP address of the virtual server that includes the pool with the encapsulation profile.

```
tmsh create ltm monitor udp npath_udp_monitor transparent enabled destination  
176.16.1.1:*
```

This command creates a transparent monitor for UDP traffic with the destination IP address `176.16.1.1`, and the port supplied by the pool member.

3. Associate the health monitor with the pool that has the encapsulation profile.

```
tmsh modify pool npath_ipip_pool monitor npath_udp_monitor
```

This command specifies that the BIG-IP® system monitors UDP traffic to the pool `npath_ipip_pool`.

4. Enable the variable in the `db` component that causes the monitor traffic to be encapsulated.

```
tmsh modify sys db tm.monitorencap value enable
```

This command specifies that the monitor traffic is encapsulated.

Layer 3 nPath routing example

The following illustration shows one example of an L3 nPath routing configuration in a network.

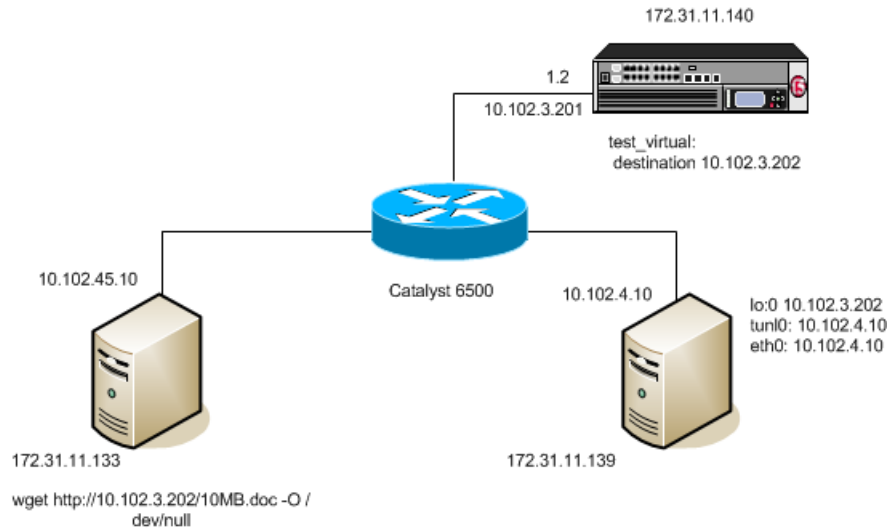


Figure 5: Example of a Layer 3 routing configuration

The following examples show the configuration code that supports the illustration.

Client configuration:

```
# ifconfig eth0 inet 10.102.45.10 netmask 255.255.255.0 up
# route add -net 10.0.0.0 netmask 255.0.0.0 gw 10.102.45.1
```

BIG-IP® device configuration:

```
# - create node pointing to server's ethernet address
# ltm node 10.102.4.10 {
#   address 10.102.4.10
# }
# - create transparent monitor
# ltm monitor tcp t.ipip {
#   defaults-from tcp
#   destination 10.102.3.202:http
#   interval 5
#   time-until-up 0
#   timeout 16
#   transparent enabled
# }
# - create pool with ipip profile
```

```
# ltm pool ipip.pool {
#   members {
#     10.102.4.10:any {           - real server's ip address
#       address 10.102.4.10
#     }
#   }
#   monitor t.ipip               - transparent monitor
#   profiles {
#     ipip
#   }
# }
# - create FastL4 profile with PVA disabled
# ltm profile fastl4 fastL4.ipip {
#   app-service none
#   pva-acceleration none
# }
# - create FastL4 virtual with custom FastL4 profile from previous step
# ltm virtual test_virtual {
#   destination 10.102.3.202:any - server's loopback address
#   ip-protocol tcp
#   mask 255.255.255.255
#   pool ipip.pool              - pool with ipip profile
#   profiles {
#     fastL4.ipip { }           - custom fastL4 profile
#   }
#   translate-address disabled  - translate address disabled
#   translate-port disabled
#   vlans-disabled
# }
```

Linux DSR server configuration:

```
# modprobe ipip
# ifconfig tunl0 10.102.4.10 netmask 255.255.255.0 up
# ifconfig lo:0 10.102.3.202 netmask 255.255.255.255 -arp up
# echo 1 > /proc/sys/net/ipv4/conf/all/arp_ignore
# echo 2 > /proc/sys/net/ipv4/conf/all/arp_announce
# echo 0 > /proc/sys/net/ipv4/conf/tunl0/rp_filter
```

Creating a Basic Web Site and E-commerce Configuration

Overview: Basic web site and eCommerce configuration

The most common use for the BIG-IP® system is distributing traffic across an array of web servers that host standard web traffic, including eCommerce traffic. The following illustration shows a configuration where a BIG-IP system load balances two sites: `www.siterequest.com` and `store.siterequest.com`. The `www.siterequest.com` site provides standard web content, and the `store.siterequest.com` site is the e-commerce site that sells items to `www.siterequest.com` customers.

Illustration of basic web site and eCommerce configuration

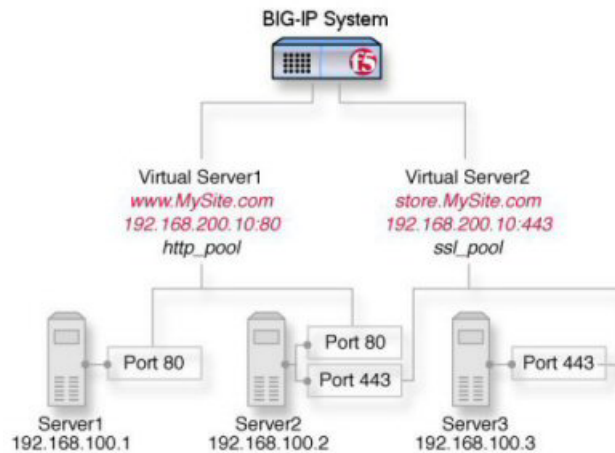


Figure 6: Basic web site and eCommerce configuration

Task summary

You can implement a basic configuration for load balancing application traffic to a web site, as well as load balancing secure traffic to an eCommerce site.

Before you use this implementation:

- Verify that you have created two VLANs on the BIG-IP® system. One VLAN should reside on the external network and another on the internal network.
- Verify that you have created a self IP address for each VLAN.

Task list

Creating a pool to process HTTP traffic

Creating a pool to manage HTTPS traffic

Creating a virtual server to manage HTTP traffic

Creating a virtual server to manage HTTPS traffic

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a pool to manage HTTPS traffic

You can create a pool (a logical set of devices, such as web servers, that you group together to receive and process HTTPS traffic) to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Assign the **https** or **https_443** health monitor from the **Available** list by moving it to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:

- Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Add each resource that you want to include in the pool using the **New Members** setting:
 - a) Type an IP address in the **Address** field.
 - b) Type 443 in the **Service Port** field, or select **HTTPS** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
 8. Click **Finished**.

The HTTPS load balancing pool now appears in the Pool List screen.

Creating a virtual server to manage HTTP traffic

You can create a virtual server to manage HTTP traffic as either a host virtual server or a network virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. From the **HTTP Compression Profile** list, select one of the following profiles:
 - **httpcompression**
 - **wan-optimized-compression**
 - A customized profile
8. (Optional) From the **Web Acceleration Profile** list, select one of the following profiles:
 - **optimized-acceleration**
 - **optimized-caching**
 - **webacceleration**
 - A customized profile
9. From the **Web Acceleration Profile** list, select one of the following profiles with an enabled application:
 - **optimized-acceleration**
 - **optimized-caching**
 - **webacceleration**
 - A customized profile
10. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
11. Click **Finished**.

The HTTP virtual server appears in the list of existing virtual servers on the Virtual Server List screen.

Creating a virtual server to manage HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. Type 443 in the **Service Port** field, or select **HTTPS** in the list.
6. Select **http** in the **HTTP Profile** list.
7. From the **HTTP Compression Profile** list, select one of the following profiles:
 - **httpcompression**
 - **wan-optimized-compression**
 - A customized profile
8. From the **Web Acceleration Profile** list, select one of the following profiles:
 - **optimized-acceleration**
 - **optimized-caching**
 - **webacceleration**
 - A customized profile
9. For the **SSL Profile (Client)** setting, from the **Available** list, select **clientssl**, and using the Move button, move the name to the **Selected** list.
10. Click **Finished**.

The HTTPS virtual server appears in the Virtual Server List screen.

Installing a BIG-IP System Without Changing the IP Network

Overview: Installing a BIG-IP system without changing the IP network

A combination of several features of the BIG-IP[®] system makes it possible for you to place a BIG-IP system in a network without changing the existing IP network. The following illustration shows the data center topology before you add the BIG-IP system. The data center has one LAN, with one IP network, 10.0.0.0. The data center has one router to the Internet, two web servers, and a back-end mail server.

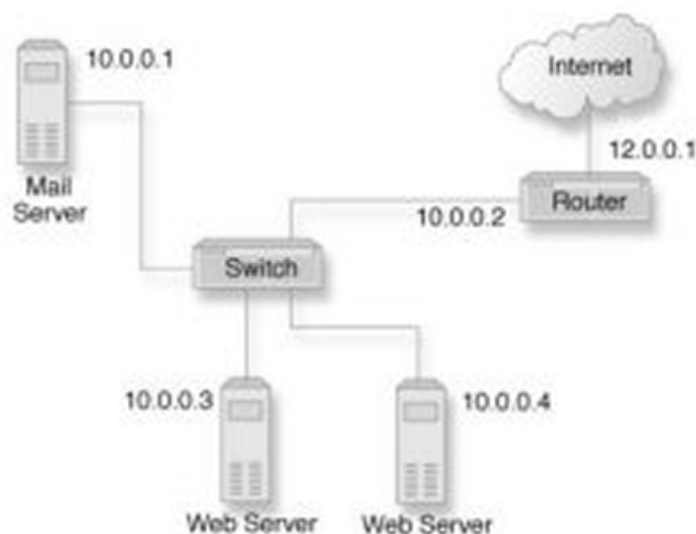


Figure 7: Data center example before adding a BIG-IP system

The existing data center structure does not support load balancing or high availability. The following illustration shows an example of the data center topology after you add the BIG-IP system.

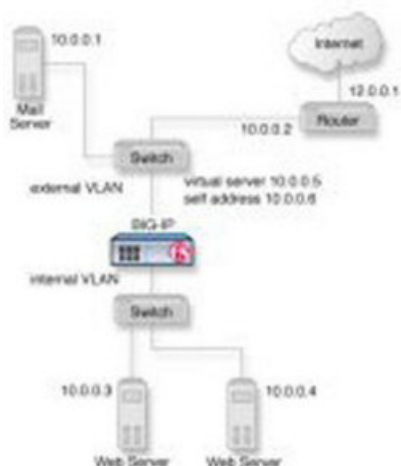


Figure 8: Data center example after adding a BIG-IP system

Task summary

To configure the BIG-IP® system for this implementation, you must perform a few key tasks. The example shown in the illustration is based on the use of the default internal and external VLAN configuration with self IP addresses on each of the VLANs that are on the same IP network on which you are installing the BIG-IP system.

Important: The default route on each content server should be set to the IP address of the router. In this example, you set the default route to **10.0.0.2**.

Task list

Removing the self IP addresses from the default VLANs

Creating a VLAN group

Creating a self IP for a VLAN group

Creating a pool of web servers

Creating a virtual server

Removing the self IP addresses from the default VLANs

Remove the self IP addresses from the individual VLANs. After you create the VLAN group, you will create another self IP address for the VLAN group for routing purposes. The individual VLANs no longer need their own self IP addresses.

1. On the Main tab, click **Network > Self IPs**.
2. Select the check box for each IP address and VLAN that you want to delete.
3. Click **Delete**.
4. Click **Delete**.

The self IP address is removed from the Self IP list.

Creating a VLAN group

VLAN groups consolidate Layer 2 traffic from two or more separate VLANs.

1. On the Main tab, click **Network > VLANs > VLAN Groups**.
The VLAN Groups list screen opens.
2. From the VLAN Groups menu, choose List.
3. Click **Create**.
The New VLAN Group screen opens.
4. In the General Properties area, in the **VLAN Group** field, type a unique name for the VLAN group.
5. For the **VLANs** setting, from the **Available** field select the **internal** and **external** VLAN names, and click << to move the VLAN names to the **Members** field.
6. Click **Finished**.

Creating a self IP for a VLAN group

A self IP address enables the BIG-IP® system and other devices on the network to route application traffic through the associated VLAN or VLAN group.

1. On the Main tab, click **Network > Self IPs**.
2. Click **Create**.
The New Self IP screen opens.
3. In the **IP Address** field, type a self IP address for the VLAN group. In the example shown, this IP address is **10.0.0.6**.
4. In the **Netmask** field, type the full network mask for the specified IP address.

For example, you can type `ffff:ffff:ffff:ffff:0000:0000:0000:0000` or `ffff:ffff:ffff:ffff::`.
5. From the **VLAN/Tunnel** list, select the name of the VLAN group you previously created.
6. From the **Port Lockdown** list, select **Allow Default**.
7. Click **Finished**.
The screen refreshes, and displays the new self IP address.

The BIG-IP system can send and receive traffic through the specified VLAN or VLAN group.

Creating a pool of web servers

You can create a pool of web servers that you group together to receive and process traffic, to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. In the Resources area of the screen, use the **New Members** setting to add the pool members. In our example, pool members are **10.0.0.3:80** and **10.0.0.4:80**.
5. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Destination** field, verify that the type of virtual server is **Host**, and in the **Address** field, type an IP address. Continuing with our example, this address would be **10.0.0.5**.

5. From the **Service Port** list, select ***All Ports**.
6. In the Resources area of the screen, from the **Default Pool** list, select a pool name.

You now have a destination IP address on the BIG-IP® system for application traffic.

Enabling IP Address Intelligence

Overview: Enabling IP address intelligence

An *IP intelligence database* is a list of IP addresses with questionable reputations. IP addresses gain a questionable reputation and are added to the database as a result of having performed exploits or attacks, or these addresses might represent proxy servers, scanners, or systems that have been infected. You can prevent system attacks by excluding traffic from malicious IP addresses. The IP Intelligence database is maintained online by a third party.

The BIG-IP® system can connect to an IP intelligence database, download the contents, and automatically keep the database up to date. You use iRules® to instruct the system on how to use IP address intelligence information. For example, iRules can instruct the system to verify the reputation of and log the originating IP address of all requests.

You can also use the IP address intelligence information within security policies in the Application Security Manager™ to log or block requests from IP addresses with questionable reputations.

Task Summary

Enabling IP address intelligence

Creating an iRule to log IP address intelligence information

Creating an iRule to reject requests with questionable IP addresses

Checking the reputation of an IP address

Checking the status of the IP intelligence database

Enabling IP address intelligence

The requirements for using IP address intelligence are:

- The system must have an IP Intelligence license.
- The system must have an Internet connection either directly or through an HTTP proxy server.
- The system must have DNS configured (go to **System > Configuration > Device > DNS**).

Important: *IP address intelligence is enabled by default. You only need to enable it if it was previously disabled.*

To enable IP address intelligence on the BIG-IP® system, you enable auto-update to connect the system to the IP intelligence database.

1. Log in to the command line for the BIG-IP® system.
2. To determine whether IP intelligence is enabled, type the following command: `tmsh list sys db iprep.autoupdate`
If the value of the `iprep.autoupdate` variable is `disable`, IP intelligence is not enabled. If it is `enable`, your task is complete.
3. At the prompt, type `tmsh modify sys db iprep.autoupdate value enable`
The system downloads the IP intelligence database and stores it in the binary file, `/var/IpRep/F5IpRep.dat`. It is updated every 5 minutes.

4. If the BIG-IP system is behind a firewall, make sure that the BIG-IP system has external access to `vector.brightcloud.com` using port 443.
That is the IP Intelligence server from which the system gets IP Intelligence information.
5. (Optional) If the BIG-IP system connects to the Internet using a forward proxy server, set these system database variables.
 - a) Type `tmsh modify sys db proxy.host value hostname` to specify the host name of the proxy server.
 - b) Type `tmsh modify sys db proxy.port value port_number` to specify the port number of the proxy server.
 - c) Type `tmsh modify sys db proxy.username value username` to specify the user name to log in to the proxy server.
 - d) Type `tmsh modify sys db proxy.password value password` to specify the password to log in to the proxy server.

The IP address intelligence feature remains enabled unless you disable it with the command `tmsh modify sys db iprep.autoupdate value disable`.

You can create iRules[®] to instruct the system how to handle traffic from IP addresses with questionable reputations, or use Application Security Manager[™] to configure IP address intelligence blocking. You can configure IP intelligence for Advanced Firewall Manager by assigning IP intelligence policies to the global, route domain, or virtual server context.

Creating an iRule to log IP address intelligence information

Before you can create an iRule to log IP address intelligence information, your system must have IP address intelligence enabled.

You use iRules[®] to log IP address intelligence categories to the file `/var/log/ltm`. This is an example of the type of iRule you can write.

1. On the Main tab, click **Local Traffic > iRules**.
The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the iRule using Tool Command Language (Tcl) syntax.
For example, to log all IP addresses and any associated IP address intelligence categories, type the following iRule:

```
when CLIENT_ACCEPTED {  
    log local0. "IP Address Intelligence for IP address  
[IP::client_addr]:  
    [IP::reputation [IP::client_addr]]"  
}
```

Tip: For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site (<http://devcentral.f5.com>).

5. Click **Finished**.
The new iRule appears in the list of iRules on the system.

When traffic is received from an IP address with a questionable reputation and that is included in the IP intelligence database, the system prints the IP address intelligence information in the `/var/log/ltm` log.

For complete and detailed information about iRules syntax, see the F5 Networks DevCentral web site, <http://devcentral.f5.com>.

Creating an iRule to reject requests with questionable IP addresses

Before you can create an iRule to reject requests based on an IP address reputation, your system must have IP address intelligence enabled.

You can use iRules® to reject requests from IP addresses that have questionable reputations and are listed in the IP intelligence database. This is an example of the type of iRule you can write.

1. On the Main tab, click **Local Traffic > iRules**.
The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the iRule using Tool Command Language (Tcl) syntax.

For example, to reject requests from IP addresses listed in the IP intelligence database because they could be Windows Exploits or Web Attacks, type the following iRule:

```
when HTTP_REQUEST {
    set ip_reputation_categories [IP::reputation [IP::client_addr]]
    set is_reject 0
    if {($ip_reputation_categories contains "Windows Exploits")} {
        set is_reject 1
    }
    if {($ip_reputation_categories contains "Web Attacks")} {
        set is_reject 1
    }
    if {($is_reject)} {
        log local0. "Attempted access from malicious IP address
[IP::client_addr]
($ip_reputation_categories), request was rejected"
        HTTP::respond 200 content
        "<HTML><HEAD><TITLE>Rejected Request</TITLE>
</HEAD><BODY>The request was rejected. <BR>
Attempted access from malicious IP address</BODY></HTML>"
    }
}
```

Tip: For complete and detailed information about iRules syntax, see the F5 Networks DevCentral web site (<http://devcentral.f5.com>).

5. Click **Finished**.
The new iRule appears in the list of iRules on the system.

When the system receives traffic from an IP address that is included in the IP intelligence database, the system prints the IP address intelligence information in the `/var/log/ltm` log.

Checking the reputation of an IP address

Before you can verify the reputation of an IP address, your system must have IP address intelligence enabled.

You can verify the reputation of a specific IP address.

1. Log in to the command line for the BIG-IP® system.
2. At the prompt, type `iprep_lookup IP_address`
where `IP_address` is the address whose reputation you want to verify. For example, to verify `1.1.1.1`:

```
iprep_lookup 1.1.1.1
opening database in /var/IpRep/F5IpRep.dat
size of IP reputation database = 41693298
iprep threats list for ip = 1.1.1.1 is:
    bit 4 - Scanners
    bit 5 - Denial of Service
```

The system looks up the IP address, and if it is in the database, the command output displays the IP address intelligence categories that show the reason. In this case, `1.1.1.1` is a source of potential port or network scans and DoS attacks. If the IP address is not found in the IP intelligence database, the system returns the message `iprep_lookup not found for ip = <ip_address>`.

Checking the status of the IP intelligence database

You can display the status of the IP Intelligence database to learn when it was last updated and the number of questionable IP addresses it contains.

1. Log in to the command line for the BIG-IP® system.
2. To display IP intelligence database status, type `tmsh show sys iprep-status`.
The system displays the status. For example:

```
-----
Sys::IP Reputation Database Status
-----
Last time the server was contacted for updates      04/21/2012 09:33:31
Last time an update was received                   04/21/2012 09:33:31
Total number of IP Addresses in the database        5516336
Number of IP Addresses received in the last update  136
```

IP address intelligence categories

Along with the IP address, the IP intelligence database stores the category that explains the reason that the IP address is considered untrustworthy.

Category Name	Description
Botnets	IP addresses of computers that are infected with malicious software (Botnet Command and Control channels, and infected zombie machines) and are controlled as a group by a Bot master, and are now part of a botnet. Hackers can exploit botnets to send

Category Name	Description
	spam messages, launch various attacks, or cause target systems to behave in other unpredictable ways.
Cloud Provider Networks	IP addresses and networks that are used by cloud providers.
Denial-of-Service	IP addresses that have launched denial-of-service (DoS) attacks, distributed denial-of-service (DDoS) attacks, anomalous SYN flood attacks, or anomalous traffic detection. These attacks are usually requests for legitimate services, but occur at such a fast rate that targeted systems cannot respond quickly enough and become bogged down or unable to service legitimate clients.
Illegal Web sites	IP addresses that contain criminally obscene or potentially criminal internet copyright and intellectual property violations.
Infected Sources	Active IP addresses that issue HTTP requests with a low reputation index score, or that are known malicious web sites offering or distributing malware, shell code, rootkits, worms, or viruses.
Phishing	IP addresses that host phishing sites, and other kinds of fraud activities, such as ad click fraud or gaming fraud.
Proxy/Anonymous Proxies	IP addresses that are associated with web proxies that shield the originator's IP address (such as proxy and anonymization services). This category also includes TOR anonymizer addresses.
Scanners	IP addresses that are involved in reconnaissance, such as probes, host scan, domain scan, and password brute force, typically to identify vulnerabilities for later exploits.
Spam Sources	IP addresses that are known to distribute large amounts of spam email by tunneling spam messages through proxy, anomalous SMTP activities, and forum spam activities.
Web Attacks	IP addresses involved in cross site scripting, iFrame injection, SQL injection, cross domain injection, or domain password brute force.
Windows Exploits	Active IP addresses that have exercised various exploits against Windows resources by offering or distributing malware, shell code, rootkits, worms, or viruses using browsers, programs, downloaded files, scripts, or operating system vulnerabilities.

Managing Client-side HTTPS Traffic Using a Self-signed Certificate

Overview: Managing client-side HTTPS traffic using a self-signed certificate

When you want to manage HTTP traffic over SSL, you can configure the BIG-IP® system to perform the SSL handshake that target web servers typically perform.

A common way to configure the BIG-IP system is to enable client-side SSL, which makes it possible for the system to decrypt client requests before forwarding them to a server, and to encrypt server responses before returning them to the client. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.

This implementation uses a self-signed certificate to authenticate HTTPS traffic.

Task summary

To implement client-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

Creating a self-signed SSL certificate

Creating a custom HTTP profile

Creating a custom Client SSL profile

Creating a pool to process HTTP traffic

Creating a virtual server for client-side HTTPS traffic

Creating a self-signed SSL certificate

If you are configuring the BIG-IP system to manage client-side HTTP traffic, you create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System > File Management > SSL Certificate List**.
The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.

8. In the **Locality** field, type your city name.
9. In the **State** or **Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select a key type.

Possible values are: **RSA**, **DSA**, and **ECDSA**.
15. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
16. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

***Note:** Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

1. On the Main tab, click **Local Traffic > Profiles > Services > HTTP**.

The HTTP profile list screen opens.
2. Click **Create**.

The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of:

- Authenticating and decrypting ingress client-side SSL traffic
- Re-encrypting egress client-side traffic

By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.

The Client profile list screen opens.
2. Click **Create**.

The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
The settings become available for change.
7. Select the **Custom** check box for **Client Authentication**.
The settings become available.
8. From the **Configuration** list, select **Advanced**.
9. Modify the settings, as required.
10. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.

2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 443, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

The HTTPS virtual server appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system can authenticate and decrypt HTTPS traffic coming from a client system. The BIG-IP system can also re-encrypt server responses before sending them back to the client.

Managing Client and Server HTTPS Traffic using a Self-signed Certificate

Overview: Managing client and server HTTPS traffic using a self-signed certificate

One of the ways to configure the BIG-IP system to manage SSL traffic is to enable both client-side and server-side SSL termination:

- *Client-side SSL termination* makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. This ensures that client-side HTTPS traffic is encrypted. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.
- *Server-side SSL termination* makes it possible for the system to decrypt and then re-encrypt client requests before sending them on to a server. Server-side SSL termination also decrypts server responses and then re-encrypts them before sending them back to the client. This ensures security for both client- and server-side HTTPS traffic. In this case, you need to install two SSL key/certificate pairs on the BIG-IP system. The system uses the first certificate/key pair to authenticate the client, and uses the second pair to request authentication from the server.

This implementation uses a self-signed certificate to authenticate HTTPS traffic.

Task summary

To implement client-side and server-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

Creating a self-signed SSL certificate

Creating a custom HTTP profile

Creating a custom Client SSL profile

Creating a custom Server SSL profile

Creating a pool to manage HTTPS traffic

Creating a virtual server for client-side and server-side HTTPS traffic

Creating a self-signed SSL certificate

If you are configuring the BIG-IP system to manage client-side HTTP traffic, you create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System > File Management > SSL Certificate List**.

The SSL Certificate List screen opens.

2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the **State** or **Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select a key type.

Possible values are: **RSA**, **DSA**, and **ECDSA**.
15. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
16. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP[®] system to manage HTTP traffic.

***Note:** Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.

The HTTP profile list screen opens.
2. Click **Create**.

The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP[®] system to terminate client-side SSL traffic for the purpose of:

- Authenticating and decrypting ingress client-side SSL traffic
- Re-encrypting egress client-side traffic

By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
The settings become available for change.
7. Select the **Custom** check box for **Client Authentication**.
The settings become available.
8. From the **Configuration** list, select **Advanced**.
9. Modify the settings, as required.
10. Click **Finished**.

Creating a custom Server SSL profile

With an Server SSL profile, the BIG-IP® system can perform decryption and encryption for server-side SSL traffic.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Server**.
The SSL Server profile list screen opens.
2. Click **Create**.
The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
The settings become available for change.
7. Select the **Custom** check box for **Server Authentication**.
8. Modify the settings, as required.
9. Click **Finished**.

The custom Server SSL profile is listed in the **Profiles:SSL:Server** list.

Creating a pool to manage HTTPS traffic

You can create a pool (a logical set of devices, such as web servers, that you group together to receive and process HTTPS traffic) to efficiently distribute the load on your server resources.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.

3. In the **Name** field, type a unique name for the pool.
4. Assign the **https** or **https_443** health monitor from the **Available** list by moving it to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Add each resource that you want to include in the pool using the **New Members** setting:
 - a) Type an IP address in the **Address** field.
 - b) Type 443 in the **Service Port** field, or select **HTTPS** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The HTTPS load balancing pool now appears in the Pool List screen.

Creating a virtual server for client-side and server-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. Type 443 in the **Service Port** field, or select **HTTPS** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
9. Click **Finished**.

The HTTPS virtual server now appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system ensures that SSL authentication and encryption occurs for both client-side and server-side HTTP traffic. The system performs this authentication and encryption according to the values you specify in the Client SSL and Server SSL profiles.

Securing HTTP Traffic Using a Self-signed Certificate with an Elliptic Curve DSA Key

Overview: Managing client-side HTTP traffic using a self-signed, ECC-based certificate

When you configure the BIG-IP[®] system to decrypt client-side HTTP requests and encrypt the server responses, you can optionally configure the BIG-IP system to use an Elliptic Curve Digital Signature Algorithm (ECDSA) key for authentication as part of the BIG-IP system's certificate key chain. Using elliptic curve cryptography (ECC), an ECDSA key creates a digital signature that allows the system to verify the authenticity of data without compromising its security. The result is that the BIG-IP system performs the SSL handshake, usually performed by target web servers, using an ECDSA key type in the certificate key chain.

This particular implementation uses a self-signed certificate.

Task summary

To implement client-side authentication using HTTP and SSL with a self-signed certificate, you perform a few basic configuration tasks.

Task list

- Creating a self-signed SSL certificate*
- Creating a custom HTTP profile*
- Creating a custom Client SSL profile*
- Creating a pool to process HTTP traffic*
- Creating a virtual server for client-side HTTPS traffic*

Creating a self-signed SSL certificate

If you are configuring the BIG-IP system to manage client-side HTTP traffic, you create a self-signed certificate to authenticate and secure the client-side HTTP traffic. If you are also configuring the system to manage server-side HTTP traffic, you create a second self-signed certificate to authenticate and secure the server-side HTTP traffic.

1. On the Main tab, click **System > File Management > SSL Certificate List**.
The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Self**.
5. In the **Common Name** field, type a name.

6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the **State** or **Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.
14. From the **Key Type** list, select **ECDSA**.
15. From the **Curve Name** list, select a curve name.

Possible values are **prime256v1** and **secp384r1**.
16. Click **Finished**.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

***Note:** Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

1. On the Main tab, click **Local Traffic > Profiles > Services > HTTP**.

The HTTP profile list screen opens.
2. Click **Create**.

The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server. When you perform this task, the system uses a certificate key chain that specifies Elliptic Curve Digital Signature Algorithm (ECDSA) as the key type.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.

The Client profile list screen opens.
2. Click **Create**.

The New Client SSL Profile screen opens.

3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
 - a) From the **Certificate** list, select the name of a certificate with a key of type ECDSA.
 - b) From the **Key** list, select the name of an ECDSA key.
 - c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
 - d) Click **Add**.
7. Configure all other settings as needed.
8. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type **80** in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 443, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

The HTTPS virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system authenticates and encrypts client-side ingress HTTP traffic using an SSL certificate key chain. The BIG-IP system also re-encrypts server responses before sending the responses back to the client.

The certificate in the certificate key chain includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key as the authentication mechanism.

Managing Client-side HTTPS Traffic using a CA-signed Certificate

Overview: Managing client-side HTTPS traffic using a CA-signed certificate

When you want to manage HTTP traffic over SSL, you can configure the BIG-IP[®] system to perform the SSL handshake that target web servers normally perform.

A common way to configure the BIG-IP system is to enable client-side SSL, which makes it possible for the system to decrypt client requests before sending them on to a server, and encrypt server responses before sending them back to the client. In this case, you need to install only one SSL key/certificate pair on the BIG-IP system.

This implementation uses a certificate signed by a certificate authority (CA) to authenticate HTTPS traffic.

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

Requesting a certificate from a certificate authority

Creating a custom HTTP profile

Creating a custom Client SSL profile

Creating a pool to process HTTP traffic

Creating a virtual server for client-side HTTPS traffic

Requesting a certificate from a certificate authority

You can generate a certificate and copy it or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System > File Management > SSL Certificate List**.
The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.
8. In the **Locality** field, type your city name.
9. In the **State** or **Province** field, type your state or province name.

10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select a key type.

Possible values are: **RSA**, **DSA**, and **ECDSA**.
17. From the **Size** or **Curve Name** list, select either a size, in bits, or a curve name.
18. Click **Finished**.

The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
 - In the **Request Text** field, copy the certificate.
 - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.

The Certificate Signing Request screen displays.

The generated certificate is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

Note: Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.

The HTTP profile list screen opens.
2. Click **Create**.

The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of:

- Authenticating and decrypting ingress client-side SSL traffic
- Re-encrypting egress client-side traffic

By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. From the **Configuration** list, select **Advanced**.
6. Select the **Custom** check box.
The settings become available for change.
7. Select the **Custom** check box for **Client Authentication**.
The settings become available.
8. From the **Configuration** list, select **Advanced**.
9. Modify the settings, as required.
10. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 443, or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select the HTTP profile that you previously created.
7. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
8. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
9. Click **Finished**.

The HTTPS virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system can authenticate and decrypt HTTPS traffic coming from a client system. The BIG-IP system can also re-encrypt server responses before sending them back to the client.

Securing HTTP Traffic using a CA-signed Certificate with an Elliptic Curve DSA Key

Overview: Managing client-side HTTP traffic using a CA-signed, ECC-based certificate

When you configure the BIG-IP® system to decrypt client-side HTTP requests and encrypt the server responses, you can optionally configure the BIG-IP system to use an Elliptic Curve Digital Signature Algorithm (ECDSA) key for authentication as part of the BIG-IP system's certificate key chain. Using elliptic curve cryptography (ECC), an ECDSA key creates a digital signature that allows the system to verify the authenticity of data without compromising its security. The result is that the BIG-IP system performs the SSL handshake usually performed by target web servers, using an ECDSA key type in the certificate key chain.

This particular implementation uses a certificate signed by a certificate authority (CA).

Task summary

To implement client-side authentication using HTTP and SSL with a certificate signed by a certificate authority, you perform a few basic configuration tasks.

Task list

- Requesting a certificate from a certificate authority*
- Creating a custom HTTP profile*
- Creating a custom Client SSL profile*
- Creating a pool to process HTTP traffic*
- Creating a virtual server for client-side HTTPS traffic*

Requesting a signed certificate that includes an ECDSA key

You can generate a certificate that includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key type, and then copy it or submit it to a trusted certificate authority for signature.

1. On the Main tab, click **System > File Management > SSL Certificate List**.
The SSL Certificate List screen opens.
2. Click **Create**.
3. In the **Name** field, type a unique name for the SSL certificate.
4. From the **Issuer** list, select **Certificate Authority**.
5. In the **Common Name** field, type a name.
6. In the **Division** field, type your company name.
7. In the **Organization** field, type your department name.

8. In the **Locality** field, type your city name.
9. In the **State** or **Province** field, type your state or province name.
10. From the **Country** list, select the name of your country.
11. In the **E-mail Address** field, type your email address.
12. In the **Lifetime** field, type a number of days, or retain the default, **365**.
13. In the **Subject Alternative Name** field, type a name.

This name is embedded in the certificate for X509 extension purposes.

By assigning this name, you can protect multiple host names with a single SSL certificate.
14. In the **Challenge Password** field, type a password.
15. In the **Confirm Password** field, re-type the password you typed in the **Challenge Password** field.
16. From the **Key Type** list, select **ECDSA**.
17. From the **Curve Name** list, select a curve name.

Possible values are **prime256v1** and **secp384r1**.
18. Click **Finished**.

The Certificate Signing Request screen displays.
19. Do one of the following to download the request into a file on your system.
 - In the **Request Text** field, copy the certificate.
 - For **Request File**, click the button.
20. Follow the instructions on the relevant certificate authority web site for either pasting the copied request or attaching the generated request file.
21. Click **Finished**.

The Certificate Signing Request screen displays.

The generated certificate is submitted to a trusted certificate authority for signature.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

Note: Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.

The HTTP profile list screen opens.
2. Click **Create**.

The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a custom Client SSL profile

You create a custom Client SSL profile when you want the BIG-IP® system to terminate client-side SSL traffic for the purpose of decrypting client-side ingress traffic and decrypting client-side egress traffic. By terminating client-side SSL traffic, the BIG-IP system offloads these authentication and decryption/encryption functions from the destination server. When you perform this task, the system uses a certificate key chain that specifies Elliptic Curve Digital Signature Algorithm (ECDSA) as the key type.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. Select the **Custom** check box.
The settings become available for change.
6. Using the **Certificate Key Chain** setting, specify one or more certificate key chains:
 - a) From the **Certificate** list, select the name of a certificate with a key of type ECDSA.
 - b) From the **Key** list, select the name of an ECDSA key.
 - c) From the **Chain** list, select the chain that you want to include in the certificate key chain.
 - d) Click **Add**.
7. Configure all other settings as needed.
8. Click **Finished**.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.

- c) (Optional) Type a priority number in the **Priority** field.
- d) Click **Add**.

8. Click **Finished.**

The new pool appears in the Pools list.

Creating a virtual server for client-side HTTPS traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage HTTPS traffic.

- 1.** On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
- 2.** Click the **Create** button.
The New Virtual Server screen opens.
- 3.** In the **Name** field, type a unique name for the virtual server.
- 4.** For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
- 5.** In the **Service Port** field, type 443, or select **HTTPS** from the list.
- 6.** From the **HTTP Profile** list, select the HTTP profile that you previously created.
- 7.** For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
- 8.** In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
- 9.** Click **Finished**.

The HTTPS virtual server appears in the Virtual Server List screen.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system authenticates and encrypts client-side ingress HTTP traffic using an SSL certificate key chain. The BIG-IP system also re-encrypts server responses before sending the responses back to the client.

The certificate in the certificate key chain includes an Elliptic Curve Digital Signature Algorithm (ECDSA) key as the authentication mechanism.

Configuring Content Adaptation for HTTP Requests

Overview: Configuring HTTP Request Adaptation

This implementation describes how to configure the BIG-IP® content adaptation feature for adapting HTTP requests. With this feature, a BIG-IP virtual server can conditionally forward HTTP requests to a pool of Internet Content Adaptation Protocol (ICAP) servers for modification, before sending the request to a web server.

In this implementation, you create a standard HTTP virtual server and pool of web servers for processing client requests. The HTTP virtual server accepts each client request in the normal way, but before load balancing the request to the pool of web servers, the virtual server forwards the HTTP request to a special internal virtual server.

The *internal virtual server* receives the HTTP request from the standard virtual server, and load balances the request to a pool of ICAP servers for modification. After the ICAP server modifies the request, the BIG-IP system sends the request to the appropriate web server for processing.

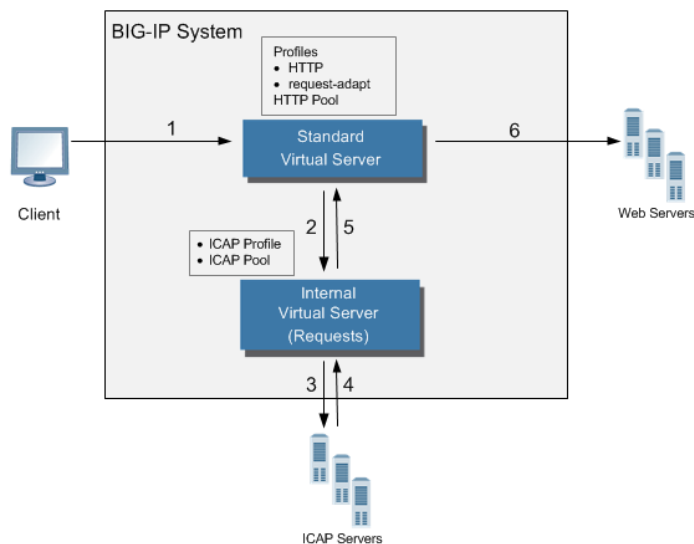


Figure 9: Content adaptation configuration for modifying HTTP requests

The internal virtual server references the pool of content adaptation servers, including the load balancing method to use for those servers. The internal virtual server also references an ICAP profile, which includes specific instructions for how the BIG-IP system should wrap the HTTP request in an ICAP message for adaptation.

Optionally, the internal virtual server can reference:

- Any persistence method that you would like the BIG-IP system to use when load balancing traffic to the ICAP pool.
- Any health or performance monitor that you would like the BIG-IP system to use when load balancing traffic to the ICAP pool.
- Any iRules® related to the content adaptation.

Task summary

Complete the tasks in this implementation to create a BIG-IP® configuration that performs content adaptation for HTTP requests.

Task List

Creating a custom client-side ICAP profile

Creating a pool of ICAP servers

Creating an internal virtual server for forwarding requests to an ICAP server

Creating a custom Request Adapt profile

Creating a custom HTTP profile

Creating a pool to process HTTP traffic

Creating an HTTP virtual server for enabling request adaptation

Creating a custom client-side ICAP profile

You create this ICAP profile when you want to use an ICAP server to wrap an HTTP request in an ICAP message before the BIG-IP® system sends the request to a pool of web servers. The profile specifies the HTTP request-header values that the ICAP server uses for the ICAP message.

Important: You can use macro expansion for all ICAP header values. For example, if an ICAP header value contains `${SERVER_IP}`, the BIG-IP system replaces the macro with the IP address of the ICAP server selected from the pool assigned to the internal virtual server. If an ICAP header value contains `${SERVER_PORT}`, the BIG-IP system replaces the macro with the port of the ICAP server selected from the pool assigned to the internal virtual server. For example, you can set the **URI** value in an ICAP profile to `icap://${SERVER_IP}:${SERVER_PORT}/virusScan`.

1. On the Main tab, click **Local Traffic > Profiles > Services > ICAP**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, `icap`.
5. On the right side of the screen, select the **Custom** check box.
6. In the **URI** field, type a URI in this format: `icap://hostname:port/path`.
For example, using macro expansion, you can set the **URI** value to:

```
icap://${SERVER_IP}:${SERVER_PORT}/virusScan
```

7. In the **Preview Length** field, type a length or retain the default value 0.
This value defines the amount of the HTTP request or response that the BIG-IP system offers to the ICAP server when sending the request or response to the server for adaptation. This value should not exceed the length of the preview that the ICAP server has indicated it will accept.
8. In the **Header From** field, type a value for the `From`: ICAP header.
9. In the **Host** field, type a value for the `Host`: ICAP header.
10. In the **Referer** field, type a value for the `Referer`: ICAP header.

11. In the **User Agent** field, type a value for the `User-Agent`: ICAP header.
12. Click **Finished**.

After you create the ICAP profile, you can assign it to an internal virtual server so that the HTTP request that the BIG-IP system sends to an ICAP server is wrapped in an ICAP message, according to the settings you specified in the ICAP profile.

Creating a pool of ICAP servers

You perform this task to create a pool of ICAP servers that perform content adaptation on HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The pool of ICAP load balancing servers appears in the Pools list.

Creating an internal virtual server for forwarding requests to an ICAP server

A virtual server of type **internal** provides a destination that a **standard** type of virtual server can use when forwarding HTTP requests slated for ICAP-based content adaptation.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Description** field, type a description of the virtual server.
For example: This virtual server ensures HTTP request modification through the use of the `service_name` ICAP service..

5. From the **Type** list, select **Internal**.
6. For the **State** setting, verify that the value is set to **Enabled**.
7. From the **Configuration** list, select **Advanced**.
8. From the **ICAP Profile** list, select the ICAP profile that you previously created for handling HTTP requests.
9. From the **Default Pool** list, select the pool of ICAP servers that you previously created.
10. Click **Finished**.

After you perform this task, a standard type of virtual server can forward HTTP requests to an internal type of virtual server. The internal virtual server then sends the request to a pool of ICAP servers, before sending the request back to the standard virtual server for forwarding to the pool of web servers.

Creating a custom Request Adapt profile

You create a Request Adapt type of profile when you want a standard HTTP virtual server to forward HTTP requests to an internal virtual server that references a pool of ICAP servers. A Request Adapt type of profile instructs the HTTP virtual server to send an HTTP request to a named internal virtual server for possible request modification.

1. On the Main tab, click **Local Traffic > Profiles > Services > Request Adapt**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, `requestadapt`.
5. On the right-side of the screen, clear the **Custom** check box.
6. For the **Enabled** setting, retain the default value, `Enabled`.
When you set this value to **Enabled**, the BIG-IP system forwards HTTP requests to the specified internal virtual server for adaptation.
7. From the **Internal Virtual Name** list, select the name of the internal virtual server that you previously created for forwarding HTTP requests to the pool of iCAP servers.
8. In the **Preview Size** field, type a numeric value.
This specifies the maximum size of the preview buffer. This buffer holds a copy of the HTTP request header and the data sent to the internal virtual server, in case the adaptation server reports that no adaptation is needed. Setting the preview size to 0 disables buffering of the request and should only be done if the adaptation server always returns a modified HTTP request or the original HTTP request.
9. In the **Timeout** field, type a numeric value, in seconds.
If the internal virtual server does not return a result within the specified time, a timeout error occurs. To disable the timeout, use the value 0.
10. From the **Service Down Action** list, select an action for the BIG-IP system to take if the internal virtual server returns an error:
 - Select **Ignore** to instruct the BIG-IP system to ignore the error and send the unmodified HTTP request to an HTTP server in the HTTP server pool.
 - Select **Drop** to instruct the BIG-IP system to drop the connection.
 - Select **Reset** to instruct the BIG-IP system to reset the connection.
11. Click **Finished**.

After you perform this task, the BIG-IP® system contains a Request Adapt profile that a standard HTTP virtual server can use to forward an HTTP request to an internal virtual server for ICAP traffic.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

Note: Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.

1. On the Main tab, click **Local Traffic > Profiles > Services > HTTP**.
The HTTP profile list screen opens.
2. Click **Create**.
The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type **80** in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating an HTTP virtual server for enabling request adaptation

You perform this task to create a standard virtual server that can forward an HTTP request to an internal virtual server. The internal virtual server then sends the request to a pool of ICAP servers before the BIG-IP® system sends the request to the web server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address that you want to use as a destination for client traffic destined for a pool of HTTP web servers.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **Configuration** list, select **Advanced**.
7. From the **HTTP Profile** list, select the name of the HTTP profile that you created previously.
8. From the **Request Adapt Profile** list, select the name of the Request Adapt profile that you previously created.
9. From the **Source Address Translation** list, select **Auto Map**.
10. From the **Default Pool** list, select the name of the HTTP server pool that you previously created.
11. Click **Finished**.

After you create the virtual server, the BIG-IP® system can forward an HTTP request to a pool of ICAP servers before sending the request to the web server.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system can perform content adaptation on HTTP requests as they pass through the BIG-IP system during normal HTTP processing. The new objects that this implementation creates are:

- A custom ICAP profile
- A pool of ICAP content adaptation servers
- An internal virtual server that load balances HTTP requests to the ICAP pool
- A custom Request Adapt profile that references the internal virtual server
- A custom HTTP profile
- A standard HTTP pool of web servers
- A standard HTTP virtual server that sends HTTP requests to an internal virtual server for content adaptation and load balances HTTP requests to the web pool

Configuring Content Adaptation for HTTP Requests and Responses

Overview: Configuring HTTP Request and Response Adaptation

This implementation describes how to configure the BIG-IP® content adaptation feature for adapting HTTP requests and responses. With this feature, a BIG-IP system virtual server can conditionally forward HTTP requests and HTTP responses to a pool of Internet Content Adaptation Protocol (ICAP) servers for modification, before sending a request to a web server or returning a response to the client system.

In this implementation, you create a standard HTTP virtual server and pool of web servers for processing client requests. The HTTP virtual server accepts each client request in the normal way, but before load balancing the request to the pool of web servers, the virtual server forwards the HTTP request to a special internal virtual server.

The *internal virtual server* receives the HTTP request from the standard virtual server, and load balances the request to a pool of ICAP servers for modification. After the ICAP server modifies the request, the BIG-IP system sends the request to the appropriate web server for processing. When the web server sends the HTTP response back to the HTTP virtual server, the BIG-IP system sends the response to a second internal virtual server, which in turn load balances the response to the pool of ICAP servers for modification. After the ICAP server modifies the response, the BIG-IP system sends the response back to the client system.

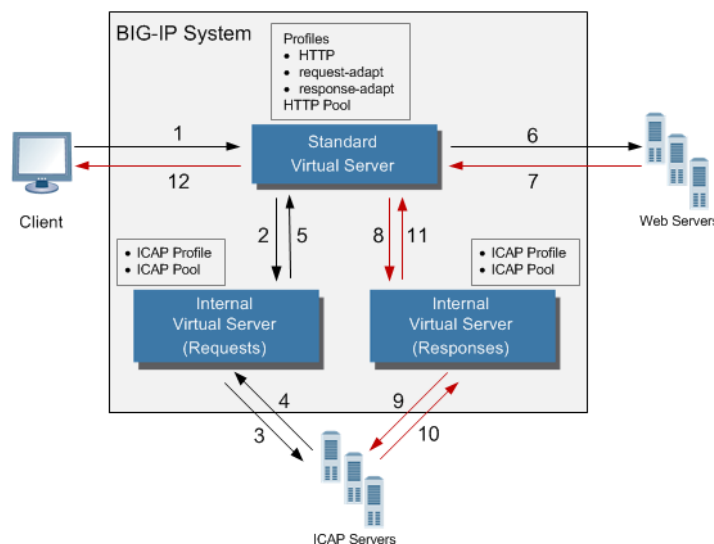


Figure 10: Content adaptation configuration for modifying HTTP requests and responses

The internal virtual server references the pool of content adaptation servers, including the load balancing method to use for those servers. The internal virtual server also references an ICAP profile, which includes specific instructions for how the BIG-IP system should modify each request or response. You can create two separate ICAP profiles, one for wrapping the HTTP request in an ICAP message for adaptation, and one for wrapping the HTTP response in an ICAP message for adaptation.

Optionally, each internal virtual server can reference:

- Any persistence method that you would like the BIG-IP system to use when load balancing traffic to the ICAP pool.
- Any health or performance monitor that you would like the BIG-IP system to use when load balancing traffic to the ICAP pool.
- Any iRules® related to the content adaptation.

Task summary

Complete the tasks in this implementation to create a BIG-IP® configuration that performs content adaptation for HTTP requests and responses.

Task List

Creating a custom client-side ICAP profile
Creating a custom server-side ICAP profile
Creating a pool of ICAP servers
Creating an internal virtual server for forwarding requests to an ICAP server
Creating an internal virtual server for forwarding responses to an ICAP server
Creating a custom Request Adapt profile
Creating a custom Response Adapt profile
Creating a custom HTTP profile
Creating a pool to process HTTP traffic
Creating an HTTP virtual server for enabling request and response adaptation

Creating a custom client-side ICAP profile

You create this ICAP profile when you want to use an ICAP server to wrap an HTTP request in an ICAP message before the BIG-IP® system sends the request to a pool of web servers. The profile specifies the HTTP request-header values that the ICAP server uses for the ICAP message.

Important: You can use macro expansion for all ICAP header values. For example, if an ICAP header value contains `${SERVER_IP}`, the BIG-IP system replaces the macro with the IP address of the ICAP server selected from the pool assigned to the internal virtual server. If an ICAP header value contains `${SERVER_PORT}`, the BIG-IP system replaces the macro with the port of the ICAP server selected from the pool assigned to the internal virtual server. For example, you can set the **URI** value in an ICAP profile to `icap://${SERVER_IP}:${SERVER_PORT}/virusScan`.

1. On the Main tab, click **Local Traffic > Profiles > Services > ICAP**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, `icap`.
5. On the right side of the screen, select the **Custom** check box.
6. In the **URI** field, type a URI in this format: `icap://hostname:port/path`.
For example, using macro expansion, you can set the **URI** value to:

```
icap://${SERVER_IP}:${SERVER_PORT}/virusScan
```

7. In the **Preview Length** field, type a length or retain the default value 0.

This value defines the amount of the HTTP request or response that the BIG-IP system offers to the ICAP server when sending the request or response to the server for adaptation. This value should not exceed the length of the preview that the ICAP server has indicated it will accept.

8. In the **Header From** field, type a value for the `From`: ICAP header.
9. In the **Host** field, type a value for the `Host`: ICAP header.
10. In the **Referer** field, type a value for the `Referer`: ICAP header.
11. In the **User Agent** field, type a value for the `User-Agent`: ICAP header.
12. Click **Finished**.

After you create the ICAP profile, you can assign it to an internal virtual server so that the HTTP request that the BIG-IP system sends to an ICAP server is wrapped in an ICAP message, according to the settings you specified in the ICAP profile.

Creating a custom server-side ICAP profile

You create this ICAP profile when you want to use an ICAP server to wrap an HTTP response in an ICAP message before the BIG-IP® system sends the response back to the client. The profile specifies the HTTP response-header values that the ICAP server uses for the ICAP message.

Important: Optionally, you can use macro expansion for all ICAP header values. For example, if an ICAP header value contains `${SERVER_IP}`, the BIG-IP system replaces the macro with the IP address of the ICAP server selected from the pool assigned to the internal virtual server. If an ICAP header value contains `${SERVER_PORT}`, the BIG-IP system replaces the macro with the port of the ICAP server selected from the pool assigned to the internal virtual server. For example, you can set the **URI** value in an ICAP profile to `icap://${SERVER_IP}:${SERVER_PORT}/videoOptimization`.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **ICAP**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, `icap`.
5. On the right side of the screen, select the **Custom** check box.
6. In the **URI** field, type a URI in this format: `icap://hostname:port/path`.
For example, using macro expansion, you can set the **URI** value to:

```
icap://${SERVER_IP}:${SERVER_PORT}/videoOptimization
```

7. In the **Preview Length** field, type a length or retain the default value 0.

This value defines the amount of the HTTP request or response that the BIG-IP system offers to the ICAP server when sending the request or response to the server for adaptation. This value should not exceed the length of the preview that the ICAP server has indicated it will accept.

8. In the **Header From** field, type a value for the `From`: ICAP header.
9. In the **Host** field, type a value for the `Host`: ICAP header.
10. In the **Referer** field, type a value for the `Referer`: ICAP header.
11. In the **User Agent** field, type a value for the `User-Agent`: ICAP header.

12. Click **Finished**.

After you create the ICAP profile, you can assign it to an internal virtual server so that the HTTP response that the BIG-IP system sends to an ICAP server is wrapped in an ICAP message, according to the settings you specified in the ICAP profile.

Creating a pool of ICAP servers

You perform this task to create a pool of ICAP servers that perform content adaptation on HTTP requests and responses.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The pool of ICAP load balancing servers appears in the Pools list.

Creating an internal virtual server for forwarding requests to an ICAP server

A virtual server of type **internal** provides a destination that a **standard** type of virtual server can use when forwarding HTTP requests slated for ICAP-based content adaptation.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Description** field, type a description of the virtual server.
For example: `This virtual server ensures HTTP request modification through the use of the service_name ICAP service..`

5. From the **Type** list, select **Internal**.
6. For the **State** setting, verify that the value is set to **Enabled**.
7. From the **Configuration** list, select **Advanced**.
8. From the **ICAP Profile** list, select the ICAP profile that you previously created for handling HTTP requests.
9. From the **Default Pool** list, select the pool of ICAP servers that you previously created.
10. Click **Finished**.

After you perform this task, a standard type of virtual server can forward HTTP requests to an internal type of virtual server. The internal virtual server then sends the request to a pool of ICAP servers, before sending the request back to the standard virtual server for forwarding to the pool of web servers.

Creating an internal virtual server for forwarding responses to an ICAP server

A virtual server of type **internal** provides a destination that a **standard** type of virtual server can use when forwarding HTTP responses slated for ICAP-based content adaptation.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. In the **Description** field, type a description of the virtual server.
For example: `This virtual server ensures HTTP response modification through the use of the service_name ICAP service..`
5. From the **Type** list, select **Internal**.
6. For the **State** setting, verify that the value is set to **Enabled**.
7. From the **Configuration** list, select **Advanced**.
8. From the **ICAP Profile** list, select the ICAP profile that you previously created for handling HTTP responses.
9. From the **Default Pool** list, select the pool of ICAP servers that you previously created.
10. Click **Finished**.

After you perform this task, a standard type of virtual server can forward an HTTP response to an internal type of virtual server. The internal virtual server then sends the response to a pool of ICAP servers before sending the response back to the standard virtual server for forwarding to the client system.

Creating a custom Request Adapt profile

You create a Request Adapt type of profile when you want a standard HTTP virtual server to forward HTTP requests to an internal virtual server that references a pool of ICAP servers. A Request Adapt type of profile instructs the HTTP virtual server to send an HTTP request to a named internal virtual server for possible request modification.

1. On the Main tab, click **Local Traffic > Profiles > Services > Request Adapt**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, `requestadapt`.

5. On the right-side of the screen, clear the **Custom** check box.
6. For the **Enabled** setting, retain the default value, *Enabled*.
When you set this value to **Enabled**, the BIG-IP system forwards HTTP requests to the specified internal virtual server for adaptation.
7. From the **Internal Virtual Name** list, select the name of the internal virtual server that you previously created for forwarding HTTP requests to the pool of iCAP servers.
8. In the **Preview Size** field, type a numeric value.
This specifies the maximum size of the preview buffer. This buffer holds a copy of the HTTP request header and the data sent to the internal virtual server, in case the adaptation server reports that no adaptation is needed. Setting the preview size to 0 disables buffering of the request and should only be done if the adaptation server always returns a modified HTTP request or the original HTTP request.
9. In the **Timeout** field, type a numeric value, in seconds.
If the internal virtual server does not return a result within the specified time, a timeout error occurs. To disable the timeout, use the value 0.
10. From the **Service Down Action** list, select an action for the BIG-IP system to take if the internal virtual server returns an error:
 - Select **Ignore** to instruct the BIG-IP system to ignore the error and send the unmodified HTTP request to an HTTP server in the HTTP server pool.
 - Select **Drop** to instruct the BIG-IP system to drop the connection.
 - Select **Reset** to instruct the BIG-IP system to reset the connection.
11. Click **Finished**.

After you perform this task, the BIG-IP® system contains a Request Adapt profile that a standard HTTP virtual server can use to forward an HTTP request to an internal virtual server for ICAP traffic.

Creating a custom Response Adapt profile

You create a Response Adapt type of profile when you want a standard HTTP virtual server to forward HTTP responses to an internal virtual server that references a pool of ICAP servers. A Response Adapt type of profile instructs the HTTP virtual server to send an HTTP response to a named internal virtual server for possible response modification.

1. On the Main tab, click **Local Traffic > Profiles > Services > Response Adapt**.
2. Click **Create**.
3. In the **Name** field, type a unique name for the profile.
4. For the **Parent Profile** setting, retain the default value, *responseadapt*.
5. On the right-side of the screen, select the **Custom** check box.
6. For the **Enabled** setting, retain the default value, *Enabled*.
When you set this value to **Enabled**, the BIG-IP system forwards HTTP responses to the specified internal virtual server for adaptation.
7. From the **Internal Virtual Name** list, select the name of the internal virtual server that you previously created for forwarding HTTP responses to the pool of iCAP servers.
8. In the **Preview Size** field, type a numeric value.
This specifies the maximum size of the preview buffer. This buffer holds a copy of the HTTP response header and the data sent to the internal virtual server, in case the adaptation server reports that no adaptation is needed. Setting the preview size to 0 disables buffering of the response and should only be done if the adaptation server always returns a modified HTTP response or the original HTTP response.
9. In the **Timeout** field, type a numeric value.

If the internal virtual server does not return a result within the specified time, a timeout error occurs. To disable the timeout, use the value 0.

10. From the **Service Down Action** list, select an action for the BIG-IP system to take if the internal virtual server returns an error:

- Select **Ignore** to instruct the BIG-IP system to ignore the error and send the unmodified HTTP response to an HTTP server in the HTTP server pool.
- Select **Drop** to instruct the BIG-IP system to drop the connection.
- Select **Reset** to instruct the BIG-IP system to reset the connection.

11. Click **Finished**.

After you perform this task, the BIG-IP® system contains a Response Adapt profile that a standard HTTP virtual server can use to forward an HTTP response to an internal virtual server for ICAP traffic.

Creating a custom HTTP profile

An HTTP profile defines the way that you want the BIG-IP® system to manage HTTP traffic.

***Note:** Other HTTP profile types (HTTP Compression and Web Acceleration) enable you to configure compression and cache settings, as required. Use of these profile types is optional.*

1. On the Main tab, click **Local Traffic > Profiles > Services > HTTP**.
The HTTP profile list screen opens.
2. Click **Create**.
The New HTTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **http**.
5. Select the **Custom** check box.
6. Modify the settings, as required.
7. Click **Finished**.

The custom HTTP profile now appears in the HTTP profile list screen.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:

- Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type **80** in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
 8. Click **Finished**.

The new pool appears in the Pools list.

Creating an HTTP virtual server for enabling request and response adaptation

You perform this task to create a standard virtual server that can forward an HTTP request or response to an internal virtual server. The internal virtual server then sends the request or response to a pool of ICAP servers before the BIG-IP[®] system sends the request or response to the client or web server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address that you want to use as a destination for client traffic destined for a pool of HTTP web servers.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type **80**, or select **HTTP** from the list.
6. From the **Configuration** list, select **Advanced**.
7. From the **HTTP Profile** list, select the name of the HTTP profile that you created previously.
8. From the **Request Adapt Profile** list, select the name of the Request Adapt profile that you previously created.
9. From the **Response Adapt Profile** list, select the name of the Response Adapt profile that you previously created.
10. From the **Source Address Translation** list, select **Auto Map**.
11. From the **Default Pool** list, select the name of the HTTP server pool that you previously created.
12. Click **Finished**.

After you create the virtual server, the BIG-IP[®] system can forward an HTTP request or response to a pool of ICAP servers before sending the request or response to the client or web server, respectively.

Implementation result

After performing the tasks in this implementation, the BIG-IP[®] can perform content adaptation on HTTP requests and responses as they pass through the BIG-IP system during normal HTTP processing. The new objects that this implementation creates are:

- Two custom ICAP profiles (for requests and responses)
- One pool of ICAP content adaptation servers
- Two separate internal virtual servers. One internal virtual server load balances HTTP requests to the ICAP pool , while the other load balances responses to the ICAP pool.
- Two custom adaptation profiles (a Request Adapt profile and a Response Adapt profile) that each reference a separate internal virtual server (for requests and responses, respectively)
- A custom HTTP profile
- A standard HTTP pool of web servers
- A standard HTTP virtual server that sends HTTP requests and responses to an internal virtual server for content adaptation, load balances HTTP requests to the web pool, and forwards HTTP responses to the relevant client

Implementing SSL Forward Proxy on a Single BIG-IP System

Overview: SSL forward proxy client and server authentication

With the BIG-IP[®] system's *SSL forward proxy* functionality, you can encrypt all traffic between a client and the BIG-IP system, by using one certificate, and to encrypt all traffic between the BIG-IP system and the server, by using a different certificate.

A client establishes a three-way handshake and SSL connection with the wildcard IP address of the BIG-IP system virtual server. The BIG-IP system then establishes a three-way handshake and SSL connection with the server, and receives and validates a server certificate (while maintaining the separate connection with the client). The BIG-IP system uses the server certificate to create a second unique server certificate to send to the client. The client receives the second server certificate from the BIG-IP system, but recognizes the certificate as originating directly from the server.

Important: To enable SSL forward proxy functionality, you can either:

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
- Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.

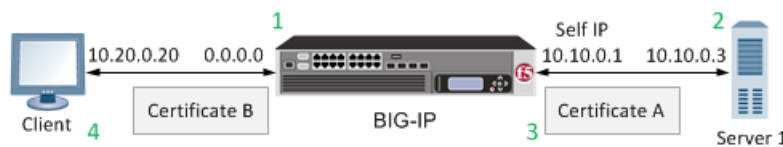


Figure 11: A virtual server configured with Client and Server SSL profiles for SSL forward proxy functionality

1. Client establishes three-way handshake and SSL connection with wildcard IP address.
2. BIG-IP system establishes three-way handshake and SSL connection with server.
3. BIG-IP system validates a server certificate (Certificate A), while maintaining the separate connection with the client.
4. BIG-IP system creates different server certificate (Certificate B) and sends it to client.

Task summary

To implement SSL forward proxy client-to-server authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the SSL Forward Proxy feature in both profiles.

Task list

Creating a custom Client SSL forward proxy profile

Creating a custom Server SSL forward proxy profile

Creating a load balancing pool

Creating a virtual server for client-side and server-side SSL traffic

Creating a custom Client SSL forward proxy profile

You perform this task to create a Client SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select **clientssl**.
5. From the **SSL Forward Proxy** list, select **Advanced**.
6. Select the **Custom** check box for the SSL Forward Proxy area.
7. Modify the SSL Forward Proxy settings.
 - a) From the **SSL Forward Proxy** list, select **Enabled**.
 - b) From the **CA Certificate** list, select a certificate.
 - c) From the **CA Key** list, select a key.
 - d) In the **CA Passphrase** field, type a passphrase.
 - e) In the **Confirm CA Passphrase** field, type the passphrase again.
 - f) In the **Certificate Lifespan** field, type a lifespan for the SSL forward proxy certificate in days.
 - g) (Optional) From the **Certificate Extensions** list, select **Extensions List**.
 - h) (Optional) For the **Certificate Extensions List** setting, select the extensions that you want in the **Available extensions** field, and move them to the **Enabled Extensions** field using the **Enable** button.
 - i) Select the **Cache Certificate by Addr-Port** check box if you want to cache certificates by IP address and port number.
 - j) From the **SSL Forward Proxy Bypass** list, select **Enabled**.
Additional settings display.
 - k) From the **Bypass Default Action** list, select **Intercept** or **Bypass**.
The default action applies to addresses and hostnames that do not match any entry specified in the lists that you specify. The system matches traffic first against destination IP address lists, then source IP address lists, and lastly, hostname lists. Within these, the default action also specifies whether to search the intercept list or the bypass list first.

Note: *If you select **Bypass** and do not specify any additional settings, you introduce a security risk to your system.*

8. Click **Finished**.

The custom Client SSL forward proxy profile now appears in the Client SSL profile list screen.

Creating a custom Server SSL forward proxy profile

You perform this task to create a Server SSL forward proxy profile that makes it possible for client and server authentication while still allowing the BIG-IP® system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL forward proxy traffic only.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Server**.
The SSL Server profile list screen opens.
2. Click **Create**.
The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list select **serverssl**.
5. Select the **Custom** check box for the Configuration area.
6. From the **SSL Forward Proxy** list, select **Enabled**.
7. Click **Finished**.

The custom Client SSL forward proxy profile now appears in the Client SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: You must create the pool before you create the corresponding virtual server.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.

8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. Specify the **Destination** settings.
 - For a Host, in the **Address** field, type 0.0.0.0 for the virtual server address.
 - For a Network, in the **Address** field, type 0.0.0.0 for the virtual server address, and in the **Mask** field, type 0.0.0.0 for the mask.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

Important: To enable SSL forward proxy functionality, you can either:

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
- Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.

7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the Server SSL forward proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

Important: To enable SSL forward proxy functionality, you can either:

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the SSL Forward Proxy settings.
- Create new Client SSL and Server SSL profiles and configure the SSL Forward Proxy settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL forward proxy functionality.

8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP® system ensures that the client system and server system can authenticate each other independently. After client and server authentication, the BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Implementing Proxy SSL on a Single BIG-IP System

Overview: Direct client-server authentication with application optimization

When setting up the BIG-IP® system to process application data, you might want the destination server to authenticate the client system directly, for security reasons, instead of relying on the BIG-IP system to perform this function. Retaining direct client-server authentication provides full transparency between the client and server systems, and grants the server final authority to allow or deny client access.

The feature that makes it possible for this direct client-server authentication is known as *Proxy SSL*. You enable this feature when you configure the Client SSL and Server SSL profiles.

Note: *To use this feature, you must configure both a Client SSL and a Server SSL profile.*

Without the Proxy SSL feature enabled, the BIG-IP system establishes separate client-side and server-side SSL connections and then manages the initial authentication of both the client and server systems.

With the Proxy SSL feature, the BIG-IP system makes it possible for direct client-server authentication by establishing a secure SSL tunnel between the client and server systems and then forwarding the SSL handshake messages from the client to the server and vice versa. After the client and server successfully authenticate each other, the BIG-IP system uses the tunnel to decrypt the application data and intelligently manipulate (optimize) the data as needed.

Task summary

To implement direct client-to-server SSL authentication, as well as application data manipulation, you perform a few basic configuration tasks. Note that you must create both a Client SSL and a Server SSL profile, and enable the Proxy SSL feature in both profiles.

Before you begin, verify that the client system, server system, and BIG-IP® system contain the appropriate SSL certificates for mutual authentication.

Important: *The BIG-IP certificate and key referenced in a Server SSL profile must match those of the server system.*

As you configure your network for Proxy SSL, keep in mind the following considerations:

- Proxy SSL supports only the RSA key exchange. For proper functioning, the client and server must not negotiate key exchanges or cipher suites that Proxy SSL does not support, such as the Diffie-Hellman (DH) and Ephemeral Diffie-Hellman (DHE) key exchanges, and the Elliptic Curve Cryptography (ECC) cipher suite. To avoid this issue, you can either configure the client so that the ClientHello packet does not include DH, DHE, or ECC; or configure the server to not accept DH, DHE, or ECC.
- Proxy SSL supports only the NULL compression method.

Creating a custom Server SSL profile

Creating a custom Client SSL profile

Creating a load balancing pool

Creating a virtual server for client-side and server-side SSL traffic

Creating a custom Server SSL profile

You perform this task to create a Server SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP[®] system to perform data optimization, such as decryption and encryption. This profile applies to server-side SSL traffic only.

Important: The certificate and key that you specify in this profile must match the certificate/key pair that you expect the back-end server to offer. If the back-end server has two or more certificates to offer, you must create a separate Server SSL profile for each certificate and then assign all of the Server SSL profiles to a single virtual server.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Server**.
The SSL Server profile list screen opens.
2. Click **Create**.
The New Server SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **serverssl** in the **Parent Profile** list.
5. From the **Certificate** list, select a relevant certificate name.
6. From the **Key** list, select a relevant key name.
7. For the **Proxy SSL** setting, select the check box.
8. From the **Configuration** list, select **Advanced**.
9. Modify all other settings, as required.
10. Choose one of the following actions:
 - If you need to create another Server SSL profile, click **Repeat**.
 - If you do not need to create another Server SSL profile, click **Finished**.

All relevant Server SSL profiles now appear on the SSL Server profile list screen.

Creating a custom Client SSL profile

You perform this task to create a Client SSL profile that makes it possible for direct client-server authentication while still allowing the BIG-IP system to perform data optimization, such as decryption and encryption. This profile applies to client-side SSL traffic only.

1. On the Main tab, click **Local Traffic > Profiles > SSL > Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **clientssl** in the **Parent Profile** list.
5. For the **Proxy SSL** setting, select the check box.
6. From the **Configuration** list, select **Advanced**.
7. Modify all other settings, as required.
8. Click **Finished**.

The custom Client SSL profile now appears in the Client SSL profile list screen.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: You must create the pool before you create the corresponding virtual server.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for client-side and server-side SSL traffic

You can specify a virtual server to be either a host virtual server or a network virtual server to manage application traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, select the type, and type an address, or an address and mask, as appropriate for your network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.

6. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the custom Client SSL proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

Important: To enable proxy SSL functionality, you can either:

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
- Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable proxy SSL functionality.

7. For the **SSL Profile (Server)** setting, from the **Available** list, select the name of the custom Server SSL proxy profile you previously created, and using the Move button, move the name to the **Selected** list.

Important: To enable SSL proxy functionality, you can either:

- Disassociate existing Client SSL and Server SSL profiles from a virtual server and configure the Proxy SSL settings.
- Create new Client SSL and Server SSL profiles and configure the Proxy SSL settings.

Then with either option, select the Client SSL and Server SSL profiles on a virtual server. You cannot modify existing Client SSL and Server SSL profiles while they are selected on a virtual server to enable SSL proxy functionality.

8. Assign other profiles to the virtual server if applicable.
9. In the Resources area, from the **Default Pool** list, select the name of the pool that you created previously.
10. Click **Finished**.

The virtual server now appears in the Virtual Server List screen.

Implementation result

After you complete the tasks in this implementation, the BIG-IP[®] system ensures that the client system and server system can initially authenticate each other directly. After client-server authentication, the BIG-IP system can intelligently decrypt and manipulate the application data according to the configuration settings in the profiles assigned to the virtual server.

Configuring HTTP Load Balancing with Source Address Affinity Persistence

Overview: HTTP load balancing with source affinity persistence

Many computing environments want to use a BIG-IP[®] system to intelligently manage their HTTP traffic. You can easily control your HTTP traffic by implementing a BIG-IP system feature known as an HTTP profile. An HTTP profile is a group of settings that affect the behavior of HTTP traffic. An HTTP profile defines the way that you want the BIG-IP system to manage HTTP traffic.

You can use the default HTTP profile, with all of its default values, or you can create a custom HTTP profile. This particular implementation uses the default HTTP profile.

When you configure the BIG-IP system to manage HTTP traffic, you can also implement simple session persistence, also known as *source address affinity persistence*. Source address affinity persistence directs session requests to the same server based solely on the source IP address of a packet. To implement source address affinity persistence, the BIG-IP system offers a default persistence profile that you can implement. Just as for HTTP, you can use the default profile, or you can create a custom simple persistence profile.

Task summary

This implementation describes how to set up a basic HTTP load balancing scenario and source address affinity persistence, using the default HTTP and source address affinity persistence profiles.

Because this implementation configures HTTP load balancing and session persistence using the default HTTP and persistence profiles, you do not need to specifically configure these profiles. Instead, you simply configure some settings on the virtual server when you create it.

Task list

Creating a pool to process HTTP traffic

Creating a virtual server for HTTP traffic

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for HTTP traffic

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. From the **Default Persistence Profile** list, select **source_addr**.
This implements simple persistence, using the default source address affinity profile.
9. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Configuring HTTP Load Balancing with Cookie Persistence

Overview: HTTP load balancing with cookie persistence

Many computing environments want to use a BIG-IP® system to intelligently manage their HTTP traffic. You can easily control your HTTP traffic by implementing a BIG-IP system feature known as an HTTP profile. An HTTP profile is a group of settings that affects the behavior of HTTP traffic. An HTTP profile defines the way that you want the system to manage HTTP traffic.

You can use the default HTTP profile, with all of its default values, or you can create a custom HTTP profile. When you create a custom HTTP profile, you not only modify the setting values, but you can enable more advanced features such as data compression of server responses.

When you configure the BIG-IP system to manage HTTP traffic, you can also implement cookie-based session persistence. *Cookie persistence* directs session requests to the same server based on HTTP cookies that the BIG-IP system stores in the client's browser.

Task summary

This implementation describes how to set up a basic HTTP load balancing scenario and cookie persistence, using the default HTTP profile.

Because this implementation configures HTTP load balancing and session persistence using the default HTTP, you do not need to specifically configure this profile. Instead, you simply configure some settings on the virtual server when you create it.

Task list

- Creating a custom cookie persistence profile*
- Creating a pool to process HTTP traffic*
- Creating a virtual server for HTTP traffic*

Creating a custom cookie persistence profile

A good way to implement cookie persistence is to create a custom cookie persistence profile.

1. On the Main tab, click **Local Traffic > Profiles > Persistence**.
The Persistence profile list screen opens.
2. Click **Create**.
The New Persistence Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Persistence Type** list, select **Cookie**.
5. From the **Parent Profile** list, select **cookie**.
6. Select the **Custom** check box.

7. From the **Cookie Method** list, select **HTTP Cookie Insert**.
8. Clear the **Session Cookie** check box.
9. Type **60** in the **Minutes** field.
10. Click **Finished**.

The custom cookie persistence profile appears in the Persistence list.

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type **80** in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for HTTP traffic

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

Note: You can also use HTTP Cookie Insert persistence with a Performance (HTTP) type of virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.

4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. From the **Default Persistence Profile** list, select the name of the custom cookie profile you created earlier, such as `mycookie_profile`.
This implements cookie persistence, using a custom cookie persistence profile.
9. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Compressing HTTP Responses

Overview: Compressing HTTP responses

An optional feature of the BIG-IP® system is the system's ability to off-load HTTP compression tasks from the target server. All of the tasks that you need to configure HTTP compression, as well as the compression software itself, are centralized on the BIG-IP system. The primary way to enable HTTP compression is by configuring an HTTP Compression type of profile and then assigning the profile to a virtual server. This causes the system to compress HTTP content for any responses matching the values that you specify in the **Request-URI** or **Content-Type** settings of the HTTP Compression profile.

***Tip:** If you want to enable HTTP compression for specific connections, you can write an iRule that specifies the `HTTP::compress enable` command. Using the BIG-IP system HTTP compression feature, you can include or exclude certain types of URIs or files that you specify. This is useful because some URI or file types might already be compressed. F5 Networks does not recommend using CPU resources to compress already-compressed data because the cost of compressing the data usually outweighs the benefits. Examples of regular expressions that you might want to specify for exclusion are `.*\.pdf`, `.*\.gif`, or `.*\.html`.*

Task summary

To configure HTTP data compression, you need to create an HTTP compression type of profile, as well as a virtual server.

Task list

Creating a customized HTTP compression profile
Creating a virtual server for HTTP compression

Creating a customized HTTP compression profile

If you need to adjust the compression settings to optimize compression for your environment, you can modify a custom HTTP compression profile.

1. On the Main tab, click **Acceleration > Profiles > HTTP Compression**.
The HTTP Compression profile list screen opens.
2. Click **Create**.
The New HTTP Compression profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select one of the following profiles:
 - **httpcompression**.
 - **wan-optimized-compression**.
5. Select the **Custom** check box.

6. Modify the settings, as required.
7. Click **Finished**.

The modified HTTP compression profile is available in the **HTTP Compression** list screen.

Creating a virtual server for HTTP compression

You can create a virtual server that uses an HTTP profile with an HTTP compression profile to compress HTTP responses.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. Select **http** in the **HTTP Profile** list.
7. From the **HTTP Compression Profile** list, select one of the following profiles:
 - **httpcompression**
 - **wan-optimized-compression**
 - A customized profile
8. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
9. Click **Finished**.

The virtual server with an HTTP profile configured with an HTTP compression profile appears in the Virtual Server list.

After you have created a custom HTTP Compression profile and a virtual server, you can test the configuration by attempting to pass HTTP traffic through the virtual server. Check to see that the BIG-IP system includes and excludes the responses that you specified in the custom profile, and that the system compresses the data as specified.

Managing HTTP Traffic with the SPDY Profile

Overview: Managing HTTP traffic with the SPDY profile

You can use the BIG-IP® Local Traffic Manager™ SPDY (pronounced "speedy") profile to minimize latency of HTTP requests by multiplexing streams and compressing headers. When you assign a SPDY profile to an HTTP virtual server, the HTTP virtual server informs clients that a SPDY virtual server is available to respond to SPDY requests.

When a client sends an HTTP request, the HTTP virtual server, with an assigned iRule, manages the request as a standard HTTP request. It receives the request on port 80, and sends the request to the appropriate server. When the BIG-IP provides the request to the origin web server, the virtual server's assigned iRule inserts an HTTP header into the request (to inform the client that a SPDY virtual server is available to handle SPDY requests), compresses and caches it, and sends the response to the client.

A client that is enabled to use the SPDY protocol sends a SPDY request to the BIG-IP system, the SPDY virtual server receives the request on port 443, converts the SPDY request into an HTTP request, and sends the request to the appropriate server. When the server provides a response, the BIG-IP system converts the HTTP response into a SPDY response, compresses and caches it, and sends the response to the client.

Note: Source address persistence is not supported by the SPDY profile.

Summary of SPDY profile functionality

By using the SPDY profile, the BIG-IP Local Traffic Manager system provides the following functionality for SPDY requests.

Creating concurrent streams for each connection.

You can specify the maximum number of concurrent HTTP requests that are accepted on a SPDY connection. If this maximum number is exceeded, the system closes the connection.

Limiting the duration of idle connections.

You can specify the maximum duration for an idle SPDY connection. If this maximum duration is exceeded, the system closes the connection.

Enabling a virtual server to process SPDY requests.

You can configure the SPDY profile on the virtual server to receive both HTTP and SPDY traffic, or to receive only SPDY traffic, based in the activation mode you select. (Note that setting this to receive only SPDY traffic is primarily intended for troubleshooting.)

Inserting a header into the request.

You can insert a header with a specific name into the request. The default name for the header is X-SPDY.

Important: The SPDY protocol is incompatible with NTLM protocols. Do not use the SPDY protocol with NTLM protocols. For additional details regarding this limitation, please refer to the SPDY specification: <http://dev.chromium.org/spdy/spdy-authentication>.

Task summary for managing HTTP and SPDY traffic

Perform these tasks to manage HTTP and SPDY requests with the BIG-IP® Local Traffic Manager™ system.

Task list

Creating a pool to process HTTP traffic

Creating an iRule for SPDY requests

Creating a virtual server to manage HTTP traffic

Creating a SPDY profile

Creating a virtual server to manage SPDY traffic

Creating a pool to process HTTP traffic

You can create a pool of web servers to process HTTP requests.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating an iRule for SPDY requests

You can create an iRule that inserts an HTTP header into responses, enabling a virtual server to respond specifically to SPDY requests.

1. On the Main tab, click **Local Traffic > iRules**.

The iRule List screen displays a list of existing iRules®.

2. Click the **Create** button.
The New iRule screen opens.
3. In the **Name** field, type a unique name for the iRule.
4. In the **Definition** field, type an iRule to insert the SPDY header.

```
ltm rule /Common/spdy_enable {
  when HTTP_RESPONSE {
    HTTP::header insert "Alternate-Protocol" "443:npn-spdy/3"
  }
}
```

***Note:** Some browsers do not support the "Alternate-Protocol" header, and require a direct HTTPS connection to a virtual server that manages SPDY traffic using port 443.*

5. Click **Finished**.

The iRule that you created is now available.

Creating a virtual server to manage HTTP traffic

You can create a virtual server to manage HTTP traffic and initiate SPDY traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, for the **iRules** setting, from the **Available** list, select the name of the SPDY iRule that you want to assign, and using the Move button, move the name into the **Enabled** list.
8. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
9. Click **Finished**.

The HTTP virtual server is now available with the specified settings.

Creating a SPDY profile

You can create a SPDY profile for a virtual server, which responds to clients that send SPDY requests with a Next Protocol Negotiation (NPN) extension in the header.

1. On the Main tab, click **Local Traffic > Profiles > Services > SPDY**.
The SPDY profile list screen opens.

2. Click **Create**.
The New SPDY Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Configuration** list, select **Advanced**.
5. Select the **Custom** check box.
6. In the **Activation Mode** list, accept the default NPN mode.
7. In the **Concurrent Streams Per Connection** field, type the number of concurrent connections to allow on a single SPDY connection.
8. In the **Connection Idle Timeout** field, type the number of seconds that a SPDY connection is left open idly before it is closed.
9. (Optional) In the **Insert Header** list, select **Enabled** to insert a header name into the request sent to the origin web server.
10. (Optional) In the **Insert Header Name** field, type a header name to insert into the request sent to the origin web server.
11. In the Protocol Versions list, select the protocol versions that you want to enable.

Option	Description
All Versions Enabled	Enables all supported SPDY protocol versions and HTTP1.1.
Select Versions	Enables one or more specific protocol versions that you specify. For the Selected Versions setting, select a protocol entry in the Available field, and move the entry to the Selected field using the Move button.

12. In the **Priority Handling** list, select how the SPDY profile handles priorities of concurrent streams within the same connection.

Option	Description
Strict	Processes higher priority streams to completion before processing lower priority streams.
Fair	Enables higher priority streams to use more bandwidth than lower priority streams, without completely blocking the lower priority streams.

13. In the **Receive Window** field, type the flow-control size for upload streams, in KB.
14. In the **Frame Size** field, type the size of the data frames, in bytes, that the SPDY protocol sends to the client.
15. In the **Write Size** field, type the total size of combined data frames, in bytes, that the SPDY protocol sends in a single write function.
16. In the **Compression Level** field, type a compression level value from 0 (no compression) through 10 (most compression).
17. In the **Compression Window Size** field, type a size, in KB, for the compression window, where a larger number increases the compression of HTTP headers, but requires more memory.
18. Click **Finished**.

A SPDY profile is now available with the specified settings.

Creating a virtual server to manage SPDY traffic

You can create a virtual server to manage SPDY traffic.

Important: Do not use the SPDY protocol with NTLM protocols as they are incompatible. For additional details regarding this limitation, please refer to the SPDY specification:

<http://dev.chromium.org/spdy/spdy-authentication>.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 443 or select **HTTPS** from the list.
6. From the **HTTP Profile** list, select **http**.
7. From the **SPDY Profile** list, select **spdy**, or a user-defined SPDY profile.
8. From the **Default Pool** list, select a pool that is configured for a SPDY profile.
9. For the **SSL Profile (Client)** setting, from the **Available** list, select **clientssl**, and using the Move button, move the name to the **Selected** list.
10. Click **Finished**.

The SPDY virtual server is now ready to manage SPDY traffic.

Using Via Headers to Acquire Information About Intermediate Routers

Overview: Using Via headers

Via headers provide useful information about intermediate routers that can be used in network analysis and troubleshooting.

Task summary for identifying intermediate information with Via headers

Perform these tasks to identify intermediate information with Via headers.

Identifying information about intermediate proxies with Via headers

Removing Via headers from requests and responses

Identifying information about intermediate proxies with Via headers

The BIG-IP® system can include Via headers (configured in an HTTP profile) in a request, a response, or both, to identify information, such as protocols and names, for intermediate proxies that forward messages.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **HTTP**.
The HTTP profile list screen opens.
2. Click the name of a user-defined profile.
3. Select the **Custom** check box.
4. In the **Send Proxy Via Header In Request** list, do one of the following:
 - Select the **Preserve** option to include the `Via` header in the client request to the origin web server.
 - Select the **Append** option, and then type a string in the **Send Proxy Via Header Host Name** field, which is appended as a comment when sending a `Via` header in a request to an origin web server.
5. In the **Send Proxy Via Header In Response** list, do one of the following:
 - Select the **Preserve** option to include the `Via` header in the client response to the client.
 - Select the **Append** option, and then type a string in the **Send Proxy Via Header Host Name** field, which is appended as a comment when sending a `Via` header in a response to a client.
6. Click **Finished**.

The BIG-IP system is configured to use Via headers to identify protocols and intermediate proxies that forward messages.

Removing Via headers from requests and responses

Via headers are configured in an HTTP profile for requests or responses.

You can remove Via headers from requests and responses if you no longer require them to identify information about intermediate proxies.

1. On the Main tab, click **Local Traffic > Profiles > Services > HTTP**.
The HTTP profile list screen opens.
2. Click the name of a user-defined profile.
3. Select the **Custom** check box.
4. In the **Send Proxy Via Header In Request** list, select **Remove**.
5. In the **Send Proxy Via Header In Response** list, select **Remove**.
6. Click **Finished**.

The BIG-IP[®] system removes Via headers, as configured, for requests and responses.

Configuring the BIG-IP System as a Reverse Proxy Server

Overview: URI translation and HTML content modification

For environments that use web servers, you might want your websites to appear differently on the external network than on the internal network. For example, you might want the BIG-IP[®] system to send traffic destined for `http://www.siterequest.com/` to the internal server `http://appserver1.siterequest.com/` instead. Normally, this translation could cause some issues, such as the web server expecting to see a certain host name (such as for name-based virtual hosting) or the web server using the internal host name and/or path when sending a redirect to client systems. Fortunately, you can configure the BIG-IP system to solve these problems.

You can also configure the BIG-IP system to modify HTML content as needed after the system has performed the URI translation.

This implementation describes an example of URI translation and HTML content modification and then provides the tasks to implement this example.

About URI translation

You can configure the BIG-IP[®] system to perform URI translation on HTTP requests. Suppose that a company named `Siterequest` has a website `www.siterequest.com`, which has a public IP address and a registered DNS entry, and therefore can be accessed from anywhere on the Internet.

Furthermore, suppose that `Siterequest` has two application servers with private IP addresses and unregistered DNS entries, inside the company's firewall. The application servers are visible within the internal network as `appserver1.siterequest.com` and `appserver2.siterequest.com`.

Because these servers have no public DNS entries, any client system that tries to access one of these servers from outside the company network receives a `no such host` error.

As the illustration shows, you can prevent this problem by configuring the BIG-IP system to act as a reverse proxy server:

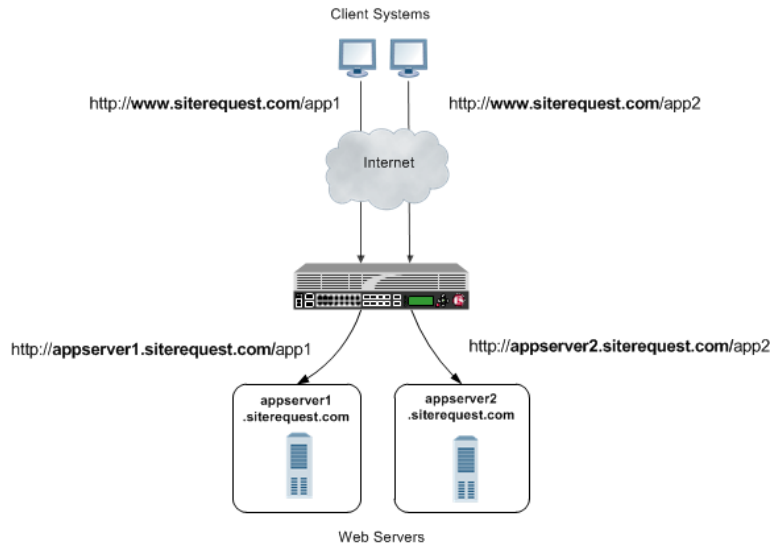


Figure 12: The BIG-IP system as a reverse proxy server for URI translation

In the example, the company *Siterequest* has decided to enable Web access to the internal application servers, without exposing them to the Internet directly. Instead, the company has integrated the servers with the web server *siterequest.com* so that `http://www.siterequest.com/sales` is mapped internally to `http://appserver1.siterequest.com/sales`, and `http://siterequest.com/marketing` is mapped internally to `http://appserver2.example.com/marketing`. This is a typical reverse-proxy configuration.

To configure the BIG-IP system to perform this translation, you create a Rewrite profile and configure one or more URI rules. A *URI rule* specifies the particular URI translation that you want the BIG-IP system to perform. Specifically, a URI rule translates the scheme, host, port, or path of any client URI, server URI, or both. A URI rule also translates any domain and path information in the *Set-Cookie* header of the response when that header information matches the information in the URI rule.

Rules for matching requests to URI rules

The BIG-IP® system follows these rules when attempting to match a request to a URI rule:

- A request does not need to match any entry. That is, if no entries match and there is no catch-all entry, then the Rewrite profile has no effect.
- Each request matches one entry only, which is the entry with the most specific host and path.
- If multiple entries match, then the BIG-IP system uses the entry with the deepest path name on the left side of the specified mapping.
- The BIG-IP system matches those requests that contain host names in URIs before matching requests that do not contain host names in URIs.
- The BIG-IP system processes the specified entries in the mapping from most-specific to least-specific, regardless of the order specified in the actual Rewrite profile.

About URI Rules

When creating a URI rule, you must specify the client and server URIs in these ways:

- When the URI is a path prefix only, the path must be preceded by and followed by a /, for example, /sales/.
- When the URI contains more than the path prefix (such as, a host), the URI must also contain a scheme and must be followed by a /, for example, http://www.siterequest/sales/.

Introduction to HTML content modification

When you configure an HTML profile on the BIG-IP® system, the system can modify HTML content that passes through the system, according to your specifications. For example, if you want the BIG-IP system to detect all content of type `text/html` and then remove all instances of the HTML `img` tag with the `src` attribute, you can configure an HTML profile accordingly, and assign it to the virtual server. The HTML profile ensures that the BIG-IP system removes those instances of the tag from any HTML content that passes through the virtual server.

Or, you can configure an HTML profile to match on a certain tag and attribute in HTML content when a particular iRule event is triggered, and then create an iRule that includes a command to replace the value of the matched attribute with a different attribute. The BIG-IP system includes several iRule commands that you can use when the `Raise Event on Comment` or `Raise Event on Tag` events are triggered. For more information on iRule commands related to HTML content modification, see the F5 Networks web site <http://devcentral.f5.com>.

HTML tag removal and replacement are just two of several HTML rules that you can configure to manipulate HTML content. An *HTML rule* defines the specific actions that you want the BIG-IP system to perform on a specified type HTML content.

Task summary

The first step to configuring the BIG-IP® system to act as a reverse proxy server is to create a Rewrite type of profile on the BIG-IP system and associate it with a virtual server. Note that each virtual server must have an HTTP profile. The Rewrite profile is designed for HTTP sites, as well as HTTPS sites where SSL is terminated on the BIG-IP system (that is, the virtual server references a Client SSL profile).

Task List

Creating a Rewrite profile to specify URI rules

Creating an HTML profile for tag removal

Creating pools for processing HTTP traffic

Creating a local traffic policy

Creating a virtual server

Creating a Rewrite profile to specify URI rules

To configure the BIG-IP® system to perform URI translation, you create a *Rewrite profile*, specifying one or more URI rules that associate a client-side path with a server-side URI. You also specify whether you want the URI translation to pertain to HTTP requests, responses, or both.

1. On the Main tab, click **Local Traffic > Profiles > Services > Rewrite**.
The Rewrite profile list appears.
2. Click **Create New Profile**.

The Create New Profile Rewrite popup screen opens.

3. In the **Profile Name** field, type a name, such as `my_rewrite_profile`.
4. From the **Parent Profile** list, select `rewrite`.
5. From the **Rewrite Mode** list, select **URI Translation**.
6. On the left pane, click **URI Rules**.
An empty text box appears for displaying client-server URI mappings that you specify.
7. Click **Add**.
8. From the **Rule Type** list, select **Both**.
9. In the **Client URI** box, type a client path, such as `/sales/`.
10. In the **Server URI** box, type a server URI, such as `http://appserver1.siterequest.com/sales/`.
You must include a scheme in the server URI that you specify.
An example of a scheme is `http`.
11. Click **OK**.
This displays a mapping of the specified client path to the associated server scheme, host, and path.
12. Click **Add** again.
13. From the **Rule Type** list, select **Both**.
14. In the **Client URI** box, type a client path, such as `/marketing/`.
15. In the **Server URI** box, type a server URI, such as
`http://appserver2.siterequest.com/marketing/`.
You must include a scheme in the server URI that you specify.
An example of a scheme is `http`.
16. Click **OK**.
This displays a mapping of the specified client path to the associated server scheme, host, and path.
17. Click **OK**.

The BIG-IP system now includes two URI rules for performing URI translation on both requests and responses. For example, the host name in a request destined for `http://www.siterequest.com/sales/` will be translated to `http://appserver1.siterequest.com/sales/`, and the host name in a request destined for `https://www.siterequest.com/marketing/` will be translated to `http://appserver2.siterequest.com/marketing/`. A reverse translation occurs on any response.

Creating an HTML profile for tag removal

You create an HTML profile when you want the BIG-IP® system to act on certain types of HTML content.

1. On the Main tab, click **Local Traffic > Profiles > Content > HTML**.
2. Click the **Create New Profile** button.
3. In the **Profile Name** field, type a name, such as `my_html_profile`.
4. From the **Parent Profile** list, select `/Common/html`.
5. On the left pane, click **HTML Rules**.
6. On the **Create New** button, click the right arrow.
7. Select **Remove Tag**.
The Create New Remove Tag Rule box appears.
8. In the **Rule Name** field, type a name, such as `my_remove_img_tag_rule`.
9. Optionally, in the **Description** field, type a description of the rule, such as `Removes the img tag with the src attribute`.

10. On the left pane, click **Match Settings**.
11. In the **Match Tag Name** field, type the name of the tag that you want to remove from the HTML content.
An example of a tag to specify is the HTML `img` tag.
12. In the **Match Attribute Name** field, type the name of the attribute associated with the tag that you specified for removal.
An example of an attribute to specify is the `src` attribute for the `img` tag.
13. Click **OK**.
14. In the **Available Rules** list, locate the HTML rule that you want to enable, and select the adjacent check box.
15. Using the Move button, move the selected HTML rule to the **Selected Rules** list.
16. Click **OK**.

After creating this HTML profile, you can implement the HTML content modification by assigning the profile to the virtual server that is processing the associated HTTP traffic.

Creating pools for processing HTTP traffic

You can create two load balancing pools, and then create a policy that forwards certain HTTP traffic to one pool, and other HTTP traffic to another pool.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
5. Click **Finished**.
6. Repeat this task to create a second pool.

The new pools appear in the Pools list.

Creating a local traffic policy

You perform this task to create a local traffic policy that forwards traffic to one or more non-default pools, based on some condition. For example, for a condition such as an HTTP request whose host name equals `siterequest.com` and URI starts with `/sales/`, the BIG-IP® system can forward that request to `pool_app1`.

1. On the Main tab, click **Local Traffic > Policies > Policy List**.
The Policy List screen opens.
2. Click **Create**.
The New Policy screen opens.
3. In the **Name** field, type a unique name for the policy.

4. From the **Strategy** list, select a matching strategy.
5. For the **Requires** setting, select **http** from the **Available** list, and move the entry to the **Selected** list using the Move button.
6. For the **Controls** setting, select **forwarding** from the **Available** list, and move the entry to the **Selected** list using the Move button.
7. Click **Add**.
The New Rule screen opens.
8. In the **Rule** field, type a unique name for the rule.
9. From the **Operand** list, select **http-host**.
10. Using the options for the **Conditions** setting, configure a rule where the condition equals the criteria specified:
 - a) From the **Condition** list, select **equals**.
 - b) (Optional) Select the **case sensitive** check box to apply case sensitivity to the condition.
 - c) In the **Values** field, type the text for the applicable value and click **Add**.
An example of a value is `siterequest.com`.
The specified condition appears in the **Values** list box.
 - d) At the lower left, click **Add**.
The configured condition appears in the **Conditions** list.
11. From the **Operand** list, select **http-uri**.
12. Using the options for the **Conditions** setting, configure a rule where the condition starts with the criteria specified:
 - a) From the **Condition** list, select **starts with**.
 - b) (Optional) Select the **case sensitive** check box to apply case sensitivity to the condition.
 - c) In the **Values** field, type the text for the applicable value and click **Add**.
An example of a value is `/app1/`.
The specified condition appears in the **Values** list box.
 - d) At the lower left click **Add**.
The configured condition appears in the **Condition** list.
13. Using the **Actions** setting, configure the applicable options:
 - a) From the **Target** list, select **forward**.
 - b) From the **Event** list, select an event.
 - c) From the **Action** list, select **pool**.
 - d) From the **Parameters** list, select the pool name to which you want the BIG-IP system to forward the traffic.
 - e) To the right of the input field, click **Add**.
The configured parameter appears in the **Parameters** list box.
 - f) At the lower left click **Add**.
The configured settings for the action appear in the **Actions** list.
14. Repeat steps 11 through 13, specifying a second `http-uri` condition value, such as `/marketing`, and specifying a different non-default pool name.
15. Click **Finished**.

For each matching condition specified in the policy, the virtual server to which you assign the policy forwards the packet to the non-default pool that you specified in the policy. For example, you can create one policy that forwards traffic with a URI starting with `/sales/` to `pool_sales` and another policy that forwards traffic with a URI starting with `/marketing/` to `pool_marketing`.

Creating a virtual server

You can create a virtual server that translates a URI in a request or response and modifies HTML content. When you create the virtual server, you can also configure it to forward certain HTTP traffic to one pool, while forwarding other HTTP traffic to a different pool..

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. For the **HTTP Profile** setting, verify that the default HTTP profile, **http**, is selected.
7. In the Content Rewrite area, from the **Rewrite Profile** list, select the relevant Rewrite profile that you created.
8. From the **HTML Profile** list, select the relevant HTML profile that you created.
9. For the **Policies** setting, from the **Available** list, select the local traffic policy you previously created, and move it to the **Enabled** list.
10. Click **Finished**.

The HTTP virtual server appears in the list of existing virtual servers on the Virtual Server List screen. This virtual server can translate URIs in requests and responses, modify HTML content, and forward the traffic to two different non-default load balancing pools.

Implementation results

After you perform the tasks in this implementation, the BIG-IP® system can:

- Translate URIs according to the URI rules specified in the Rewrite profile.
- Modify specified HTML content according to the HTML rule specified in the HTML profile.
- Forward HTTP traffic to two different non-default pools according to a local traffic policy.

Configuring the BIG-IP System as an MS SQL Database Proxy

Overview: Configuring LTM as a database proxy

You can configure BIG-IP® Local Traffic Manager™ (LTM®) systems to load balance database requests to pools of database servers. In this case, LTM acts as a proxy for databases that use the tabular data stream (TDS) protocol. LTM load balances client requests based on the user issuing the commands.

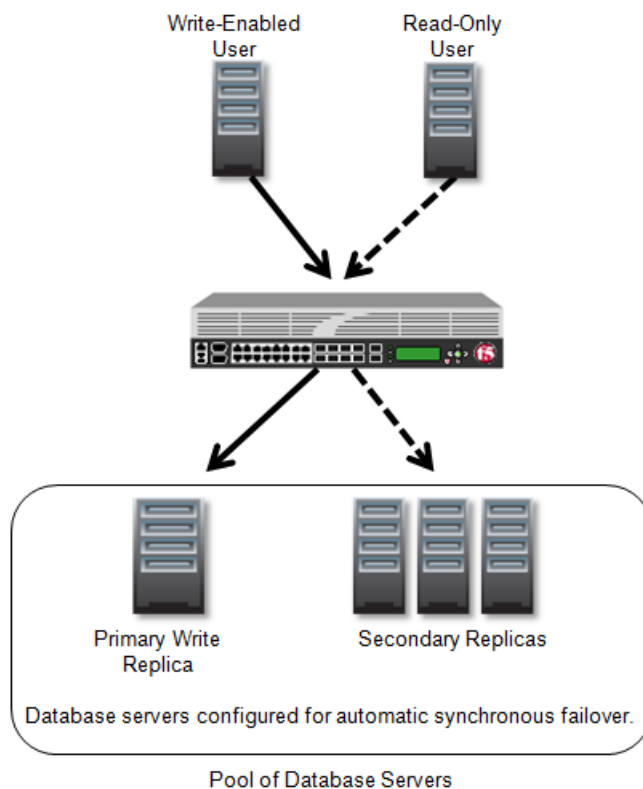


Figure 13: LTM configured as a database proxy

Task summary

About database authentication

About database access configuration

Creating a custom MS SQL monitor

Creating a pool of database servers

Configuring database access by user

Creating a custom OneConnect profile

Creating a database proxy virtual server

Viewing MS SQL profile statistics

About database authentication

BIG-IP® LTM® supports only basic authentication when acting as a proxy for an MS SQL database. You must configure user names and passwords on the database servers and the database servers must handle user authentication. Therefore, the user names and passwords must be synchronized across all database servers.

About database access configuration

You can configure BIG-IP® LTM® for user-based access to database servers. With user-based access, you configure a pool of database servers and indicate whether users write by default. Then, you configure either a read-only list of users or a write-enabled list of users.

Note: Write requests include at least one of these key words: *create, update, insert, delete, into, alter, drop, rename, exec, and execute.*

Creating a custom MS SQL monitor

Create a custom MS SQL monitor to send requests, generated using the settings you specify, to a pool of MS SQL database servers, and to validate the responses.

Important: When defining values for custom monitors, make sure you avoid using any values that are on the list of reserved keywords. For more information, see SOL number 3653 (for version 9.0 systems and later) on the AskF5™ technical support web site at www.askf5.com.

1. On the Main tab, click **Local Traffic > Monitors**.
The Monitor List screen opens.
2. Click **Create**.
The New Monitor screen opens.
3. Type a name for the monitor in the **Name** field.
4. From the **Type** list, select **MSSQL**.
5. Type a SQL statement in the **Send String** field that the monitor sends to the database server to verify availability.
This is an example of a basic Send String: `SELECT Firstname, LastName FROM Person.Person WHERE LastName = 'name'`. This is an example of a Send String that determines which database is primary: `SELECT role_desc, is_local, synchronization_health_desc FROM sys.dm_hadr_availability_replica_states WHERE is_local = 1 AND synchronization_health_desc = 'HEALTHY';`

Note: Based on the string you enter, you may need to enter values in other fields for this monitor.

6. In the **User Name** field, type the name the monitor uses to access the database server.
7. In the **Password** field, type the password the monitor uses to access the database server.
8. Click **Finished**.

Creating a pool of database servers

Gather the IP addresses of the database servers that you want to include in the pool. In an Always On architecture, normally the pool includes both primary and secondary database servers configured for synchronous automatic failover.

Ensure that a custom MS SQL monitor exists in the configuration.

Create a pool of database servers to process database requests. LTM® acts as a proxy for the database servers by load balancing requests to the members of the pool.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool of database servers.
4. For the **Health Monitors** setting, from the **Available** list, select the custom **mssql** monitor, and click << to move the monitor to the **Active** list.
5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
For pool members that are MS SQL database servers, consider **Least Connections**, which selects the server that provides the best response time.
6. Using the **New Members** setting, add the IP address for each database server that you want to include in the pool:
 - a) Type an IP address in the **Address** field, or select a node address from the **Node List**.
 - b) Type a service number in the **Service Port** field, or select a service name from the list.

***Note:** Typical TDS database servers require port 1433.*

- c) Click **Add**.
7. Click **Finished**.

The pool of database servers appears in the Pools list.

Configuring database access by user

Create a custom Microsoft SQL Server (MS SQL) profile to configure BIG-IP® LTM® to grant user-based access to a pool of database servers.

1. On the Main tab, click **Local Traffic > Profiles > Databases > MS SQL**.
The MS SQL Profiles list screen opens.
2. Click **Create**.
The New MS SQL Profile screen opens.
3. In the **Profile Name** field, type a unique name for the MS SQL profile, for example, `mssql_user_access`.
4. Select the **Custom** check box.
5. From the **Read/Write Split** list, select **By User**.
6. From the **Read Pool** list, select the pool of MS SQL database servers to which the system sends read-only requests.

7. From the **Write Pool** list, select the pool of MS SQL database servers to which the system sends write requests.
8. From the **Users Can Write By Default** list, select **Yes** to give write access to all users, except those in the **Read-Only Users** list.
9. In the **Read-Only Users** area, add users to whom you want to provide read-only access to the database.
10. Click **Finished**.

Creating a custom OneConnect profile

Optionally, you can create a custom OneConnect profile. With this profile, the LTM[®] system minimizes the number of server-side TCP connections by sharing idle connections among TDS connections owned by the same user name.

1. On the Main tab, click **Local Traffic > Profiles > Other > OneConnect**.
The OneConnect profile list screen opens.
2. Click **Create**.
The New OneConnect Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. In the Settings area, configure additional settings based on your network requirements.
5. Click **Finished**.

Creating a database proxy virtual server

Ensure that a pool of database servers exist in the configuration before creating a database proxy virtual server.

You can create a virtual server to represent a destination IP address for database transaction traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 1443.
6. From the **Configuration** list, select **Advanced**.
7. From the **MS SQL Profile** list, select either the default or a custom MS SQL profile.
8. Optionally, from the **OneConnect Profile** list, select a custom OneConnect profile.
9. From the **Default Pool** list, select the pool of database servers.

You now have a destination IP address on the BIG-IP[®] system for MS SQL database traffic.

Viewing MS SQL profile statistics

You can view statistics about database requests and responses, user access, and database messages for the traffic LTM® handles as a proxy for a database server.

1. On the Main tab, click **Statistics > Module Statistics > Local Traffic**.
The Local Traffic statistics screen opens.
2. From the **Statistics Type** list, select **Profiles Summary**.
3. In the Details column for the MS SQL profile, click **View** to display detailed statistics about database requests and responses, database access, and database messages.

Load Balancing Passive Mode FTP Traffic

Overview: FTP passive mode load balancing

You can set up the BIG-IP system to load balance passive mode FTP traffic. You do this by using the default FTP profile. An *FTP profile* determines the way that the BIG-IP system processes FTP traffic.

Additionally, you can create an iRule to apply to the FTP data channel. You apply the iRule to the data channel by assigning the iRule to the virtual server that you create.

Task Summary for load balancing passive mode FTP traffic

You can perform these tasks to configure FTP passive mode load balancing.

Task list

Creating a custom FTP monitor

Creating a pool to manage FTP traffic

Creating a virtual server for FTP traffic

Creating a custom FTP monitor

An FTP monitor requires a user name and password, and the full path to the file to be downloaded.

Create a custom FTP monitor to verify the File Transfer Protocol (FTP) service. The monitor attempts to download a specified file to the `/var/tmp` directory. If the file is retrieved, the verification is successful.

Note: The BIG-IP[®] system does not save the downloaded file.

Create a custom FTP monitor to verify passive mode File Transfer Protocol (FTP) traffic. The monitor attempts to download a specified file to the `/var/tmp` directory. If the file is retrieved, the verification is successful.

Note: The BIG-IP[®] system does not save the downloaded file.

1. On the Main tab, click **DNS > GSLB > Monitors**.
The Monitor List screen opens.
2. On the Main tab, click **Local Traffic > Monitors**.
The Monitor List screen opens.
3. Click **Create**.
The New Monitor screen opens.
4. Type a name for the monitor in the **Name** field.
5. From the **Type** list, select **FTP**.
The screen refreshes, and displays the configuration options for the **FTP** monitor type.

6. From the **Import Settings** list, select an existing monitor.
The new monitor inherits initial configuration values from the existing monitor.
7. Type a number in the **Interval** field that indicates, in seconds, how frequently the system issues the monitor check. The default is 10 seconds.
The frequency of a monitor check must be greater than the value of the global-level **Heartbeat Interval** setting. Otherwise, the monitor can acquire out-of-date data.
8. Type a number in the **Timeout** field that indicates, in seconds, how much time the target has to respond to the monitor check. The default is 31 seconds.
If the target responds within the allotted time period, it is considered up. If the target does not respond within the time period, it is considered down.
9. Type a number in the **Probe Timeout** field that indicates the number of seconds after which the system times out the probe request to the system. The default is 5 seconds.
10. Type a name in the **User Name** field.
11. Type a password in the **Password** field.
12. Type the full path and file name of the file that the system attempts to download in the **Path/Filename** field.
The health check is successful if the system can download the file.
13. For the **Mode** setting, select one of the following data transfer process (DTP) modes.

Option	Description
Passive	The monitor sends a data transfer request to the FTP server. When the FTP server receives the request, the FTP server initiates and establishes the data connection.
Port	The monitor initiates and establishes the data connection with the FTP server.
14. From the Configuration list, select **Advanced**.
This selection makes it possible for you to modify additional default settings.
15. From the **Up Interval** list, do one of the following:
 - Accept the default, **Disabled**, if you do not want to use the up interval.
 - Select **Enabled**, and specify how often you want the system to verify the health of a resource that is up.
16. Type a number in the **Time Until Up** field that indicates the number of seconds to wait after a resource first responds correctly to the monitor before setting the resource to up.
The default value is 0 (zero), which disables this option.
17. Specify whether the system automatically enables the monitored resource, when the monitor check is successful, for **Manual Resume**.
This setting applies only when the monitored resource has failed to respond to a monitor check.

Option	Description
Yes	The system does nothing when the monitor check succeeds, and you must manually enable the monitored resource.
No	The system automatically re-enables the monitored resource after the next successful monitor check.
18. For the **Alias Address** setting, do one of the following:
 - Accept the ***All Addresses** default option.

- Type an alias IP address for the monitor to verify, on behalf of the pools or pool members with which the monitor is associated.

If the health check for the alias address is successful, the system marks all associated objects up. If the health check for the alias address is not successful, then the system marks all associated objects down.

19. For the **Alias Service Port** setting, do one of the following:

- Accept the ***All Ports** default option.
- Select an alias port or service for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

If the health check for the alias port or service is successful, the system marks all associated objects up. If the health check for the alias port or service is not successful, then the system marks all associated objects down.

20. For the **Debug** setting, specify whether you want the system to collect and publish additional information and error messages for this monitor.

You can use the log information to help diagnose and troubleshoot unsuccessful health checks. To view the log entries, see the **System > Logs** screens.

Option	Description
Yes	The system redirects error messages and other information to a log file created specifically for this monitor.
No	The system does not collect additional information or error messages related to this monitor. This is the default setting.

21. Click **Finished**.

You can associate the new custom monitor with the server, virtual server, or pool member that contains the FTP resources. You can associate the new custom monitor with the pool that contains the FTP resources.

Creating a pool to manage FTP traffic

To load balance passive mode FTP traffic, you create a load balancing pool. When you create the pool, you assign the custom FTP monitor that you created in the previous task.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Priority Group Activation** list, select **Disabled**.
6. Add each resource that you want to include in the pool using the **New Members** setting:
 - a) Type an IP address in the **Address** field.
 - b) Type 21 in the **Service Port** field, or select **FTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.

7. Click **Finished**.

The pool to manage FTP traffic appears in the Pools list.

Creating a virtual server for FTP traffic

You can define a virtual server that references the FTP profile and the FTP pool.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 21 or select **FTP** from the list.
6. For the **FTP Profile** setting, select the default profile, `ftp`.
7. Locate the Resources area of the screen, and for the **Related iRules** setting, from the **Available** list, select the name of the iRule that you want to assign, and using the Move button, move the name to the **Enabled** list.
This setting applies to virtual servers that reference a profile for a data channel protocol, such as FTP or RTSP.
8. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
9. Click **Finished**.

The custom FTP virtual server appears in the Virtual Servers list.

Load Balancing Passive Mode FTP Traffic with Data Channel Optimization

Overview: FTP passive mode load balancing with data channel optimization

You can set up the BIG-IP system to load balance passive mode FTP traffic, with optimization of both the FTP control channel and the data channel.

By default, the BIG-IP system optimizes FTP traffic for the control channel, according to the configuration settings in the default client and server TCP profiles assigned to the virtual server. When you use this particular implementation, you also configure the system to take advantage of those same TCP profile settings for the FTP data channel. This provides useful optimization of the data channel payload.

Task Summary for load balancing passive mode FTP traffic

You can perform these tasks to configure FTP passive mode load balancing that optimizes traffic on both the control channel and data channel.

Task list

Creating a custom FTP profile

Creating a custom FTP monitor

Creating a pool to manage FTP traffic

Creating a virtual server for FTP traffic

Creating a custom FTP profile

You create a custom FTP profile when you want to fine-tune the way that the BIG-IP[®] system manages FTP traffic. This procedure creates an FTP profile and optimizes the way that the BIG-IP system manages traffic for the FTP data channel.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **FTP**.
The FTP profile list screen opens.
2. Click **Create**.
The New FTP Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. From the **Parent Profile** list, select the default **ftp** profile.
5. Select the **Custom** check box.
6. For the **Inherit Parent Profile** setting, select the check box.
This optimizes data channel traffic.
7. Click **Finished**.

The custom FTP profile now appears in the FTP profile list screen.

Creating a custom FTP monitor

An FTP monitor requires a user name and password, and the full path to the file to be downloaded.

Create a custom FTP monitor to verify passive mode File Transfer Protocol (FTP) traffic. The monitor attempts to download a specified file to the `/var/tmp` directory. If the file is retrieved, the check is successful.

***Note:** The BIG-IP® system does not save the downloaded file.*

1. On the Main tab, click **Local Traffic > Monitors**.
The Monitor List screen opens.
2. Click **Create**.
The New Monitor screen opens.
3. Type a name for the monitor in the **Name** field.
4. From the **Type** list, select **FTP**.
The screen refreshes, and displays the configuration options for the **FTP** monitor type.
5. From the **Import Settings** list, select an existing monitor.
The new monitor inherits initial configuration values from the existing monitor.
6. Type a number in the **Interval** field that indicates, in seconds, how frequently the system issues the monitor check. The default is 10 seconds.
The frequency of a monitor check must be greater than the value of the global-level **Heartbeat Interval** setting. Otherwise, the monitor can acquire out-of-date data.
7. Type a number in the **Timeout** field that indicates, in seconds, how much time the target has to respond to the monitor check. The default is 31 seconds.
If the target responds within the allotted time period, it is considered up. If the target does not respond within the time period, it is considered down.
8. Type a number in the **Probe Timeout** field that indicates the number of seconds after which the system times out the probe request to the system. The default is 5 seconds.
9. Type a name in the **User Name** field.
10. Type a password in the **Password** field.
11. Type the full path and file name of the file that the system attempts to download in the **Path/Filename** field.
The health check is successful if the system can download the file.

12. For the **Mode** setting, select one of the following data transfer process (DTP) modes.

Option	Description
Passive	The monitor sends a data transfer request to the FTP server. When the FTP server receives the request, the FTP server initiates and establishes the data connection.
Port	The monitor initiates and establishes the data connection with the FTP server.

13. From the Configuration list, select **Advanced**.
This selection makes it possible for you to modify additional default settings.
14. From the **Up Interval** list, do one of the following:
 - Accept the default, **Disabled**, if you do not want to use the up interval.
 - Select **Enabled**, and specify how often you want the system to verify the health of a resource that is up.

15. Type a number in the **Time Until Up** field that indicates the number of seconds to wait after a resource first responds correctly to the monitor before setting the resource to up.

The default value is 0 (zero), which disables this option.

16. Specify whether the system automatically enables the monitored resource, when the monitor check is successful, for **Manual Resume**.

This setting applies only when the monitored resource has failed to respond to a monitor check.

Option	Description
Yes	The system does nothing when the monitor check succeeds, and you must manually enable the monitored resource.
No	The system automatically re-enables the monitored resource after the next successful monitor check.

17. For the **Alias Address** setting, do one of the following:

- Accept the ***All Addresses** default option.
- Type an alias IP address for the monitor to verify, on behalf of the pools or pool members with which the monitor is associated.

If the health check for the alias address is successful, the system marks all associated objects up. If the health check for the alias address is not successful, then the system marks all associated objects down.

18. For the **Alias Service Port** setting, do one of the following:

- Accept the ***All Ports** default option.
- Select an alias port or service for the monitor to check, on behalf of the pools or pool members with which the monitor is associated.

If the health check for the alias port or service is successful, the system marks all associated objects up. If the health check for the alias port or service is not successful, then the system marks all associated objects down.

19. For the **Debug** setting, specify whether you want the system to collect and publish additional information and error messages for this monitor.

You can use the log information to help diagnose and troubleshoot unsuccessful health checks. To view the log entries, see the **System > Logs** screens.

Option	Description
Yes	The system redirects error messages and other information to a log file created specifically for this monitor.
No	The system does not collect additional information or error messages related to this monitor. This is the default setting.

20. Click **Finished**.

You can associate the new custom monitor with the pool that contains the FTP resources.

Creating a pool to manage FTP traffic

To load balance passive mode FTP traffic, you create a load balancing pool. When you create the pool, you assign the custom FTP monitor that you created in the previous task.

1. On the Main tab, click **Local Traffic > Pools**.

The Pool List screen opens.

2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

***Tip:** Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Priority Group Activation** list, select **Disabled**.
6. Add each resource that you want to include in the pool using the **New Members** setting:
 - a) Type an IP address in the **Address** field.
 - b) Type 21 in the **Service Port** field, or select **FTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
7. Click **Finished**.

The pool to manage FTP traffic appears in the Pools list.

Creating a virtual server for FTP traffic

You can define a virtual server that references the FTP profile and the FTP pool.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 21 or select **FTP** from the list.
6. From the **FTP Profile** list, select the custom profile that you created earlier.
7. Locate the Resources area of the screen, and for the **Related iRules** setting, from the **Available** list, select the name of the iRule that you want to assign, and using the Move button, move the name to the **Enabled** list.
This setting applies to virtual servers that reference a profile for a data channel protocol, such as FTP or RTSP.
8. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
9. Click **Finished**.

The custom FTP virtual server appears in the Virtual Servers list.

Implementation result

A BIG-IP system with this configuration can process FTP traffic in passive mode, in a way that optimizes the traffic on both the control channel and the data channel. This optimization is based on the settings of the default client-side and server-side TCP profiles.

Referencing an External File from within an iRule

Overview: Referencing an external file from an iRule

Using the BIG-IP® Configuration utility or **tmsh**, you can import a file or URL from another system to the BIG-IP system, with content that you want an iRule to return to a client, based on some iRule event. Possible uses for this feature are:

- To send a web page other than the page that the client requested. For example, you might want the system to send a maintenance page instead of the requested page.
- To send an image.
- To use a file as a template and modify the file in the iRule before sending the file.
- To download policy information from an external server and merge that data with a locally-stored policy.

The file that an iRule accesses is known as an *iFile*, and can be any type of file, such as a binary file or a text file. These files are read-only files.

This example shows an iRule that references an iFile named `ifileURL`, in partition `Common`:

```
ltm rule ifile_rule {
  when HTTP_RESPONSE {
    # return a list of iFiles in all partitions
    set listifiles [ifile listall]
    log local0. "list of ifiles: $listifiles"

    # return the attributes of an iFile specified
    array set array_attributes [ifile attributes "/Common/ifileURL"]
    foreach {array attr} [array get array_attributes ] {
      log local0. "$array : $attr"
    }

    # serve an iFile when http status is 404.
    set file [ifile get "/Common/ifileURL"]
    log local0. "file: $file"
    if { [HTTP::status] equals "404" } {
      HTTP::respond 200 ifile "/Common/ifileURL"
    }
  }
}
```

iRule commands for iFiles

This list shows the commands available for referencing an iFile within an iRule. All of these commands return a string, except for the command `[ifile attributes IFILENAME]`, which returns an array.

Available iRule commands for referencing an iFile

```
[ifile get IFILENAME]
```

```
[ifile listall]
[ifile attributes IFILENAME]
[ifile size IFILENAME]
[ifile last_updated_by IFILENAME]
[ifile last_update_time IFILENAME]
[ifile revision IFILENAME]
[ifile checksum IFILENAME]
[ifile attributes IFILENAME]
```

Task summary

You can import an existing file to the BIG-IP® system, create an iFile that is based on the imported file, and then write an iRule that returns the content of that file to a client system, based on an iRule event.

Task list

Importing a file to the BIG-IP system

Creating an iFile

Writing an iRule that references an iFile

Importing a file to the BIG-IP system

As a prerequisite, the file you want to import must reside on the BIG-IP® system you specify.

You can import a file from another system onto the BIG-IP system, as the first step in writing an iRule that references that file.

1. On the Main tab, click **System > File Management > iFile List > Import**.
2. For the **File Name** setting, click **Browse**.
The system opens a browse window so you can locate the file that you want to import to the BIG-IP system.
3. Browse for the file and click **Open**.
The name of the file you select appears in the **File Name** setting.
4. In the **Name** field, type a new name for the file, such as `lk.html`.
The new file name appears in the list of imported files.
5. Click **Import**.

The result of this task is that the file you selected now resides on the BIG-IP system.

Creating an iFile

As a prerequisite, ensure that the current administrative partition is set to the partition in which you want the iFile to reside.

You perform this task to create an iFile that you can then reference in an iRule.

1. On the Main tab, click **Local Traffic > iRules > iFile List**.
2. Click **Create**.
3. In the **Name** field, type a new name for the iFile, such as `ifileURL`.

4. From the **File Name** list, select the name of the imported file object, such as `lk.html`.
5. Click **Finished**.
The new iFile appears in the list of iFiles.

The result of this task is that you now have a file that an iRule can reference.

Writing an iRule that references an iFile

You perform this task to create an iRule that references an iFile.

***Note:** If the iFile resides in partition `/Common`, then specifying the partition when referencing the iFile is optional. If the iFile resides in a partition other than `/Common`, such as `/Partition_A`, you must include the partition name in the iFile path name within the iRule.*

1. On the Main tab, click **Local Traffic > iRules**.
The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.
For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site (<http://devcentral.f5.com>).
5. Click **Finished**.
The new iRule appears in the list of iRules on the system.

Implementation result

You now have an iRule that accesses a file on the BIG-IP® system, based on a particular iRule event.

Configuring the BIG-IP System as a DHCP Relay Agent

Overview: Managing IP addresses for DHCP clients

When you want to manage Dynamic Host Configuration Protocol (DHCP) client IP addresses, you can configure the BIG-IP® system to act as a DHCP relay agent. A common reason to configure the BIG-IP system as a DHCP relay agent is when the DHCP clients reside on a different subnet than the subnet of the DHCP servers.

Before configuring the BIG-IP system to act as a DHCP relay agent, it is helpful to understand some BIG-IP system terminology:

BIG-IP object type	Definition
BIG-IP pool member	A DHCP relay target (such as a DHCP server or BOOTP server). This is the dynamic address server to which the BIG-IP system forwards unicast requests.
BIG-IP virtual server	A BIG-IP system address on the listening VLAN
BIG-IP VLAN assigned to a virtual server	A listening VLAN, controlled on a per-virtual server basis

About the BIG-IP system as a DHCP relay agent

A BIG-IP® virtual server, configured as a Dynamic Host Configuration Protocol (DHCP) relay type, provides you with the ability to relay DHCP client requests for an IP address to one or more DHCP servers, available as pool members in a DHCP pool, on different virtual local area networks (VLANs). The DHCP client request is relayed to all pool members, and the replies from all pool members are relayed back to the client.

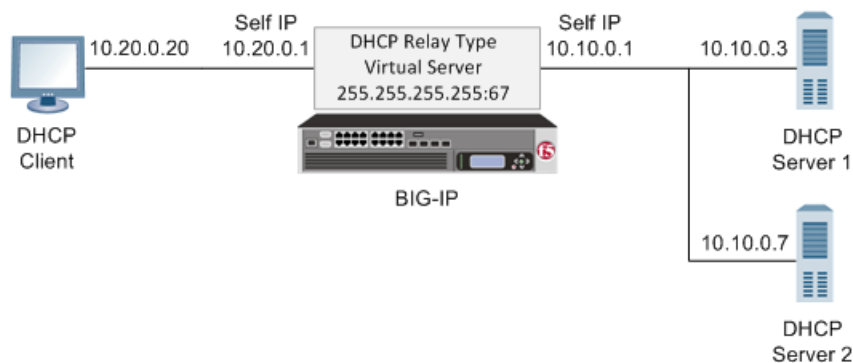


Figure 14: A sample DHCP relay agent configuration

For example, a DHCP client sends a broadcast message to the destination IP address 255.255.255.255, which is the destination address configured on the virtual server. A DHCP relay type virtual server automatically uses port 67 for an IPv4 broadcast message or port 547 for an IPv6 broadcast message. The BIG-IP virtual server receives this message on the VLAN with self IP address 10.20.0.1 and relays the DHCP request to all DHCP servers: 10.10.0.3 and 10.10.0.7.

All DHCP servers provide a DHCP response with available IP addresses to the BIG-IP virtual server, which then relays all responses to the client. The client accepts and uses only one of the IP addresses received.

Note: *In this example, there is no hop between the DHCP client and the BIG-IP relay agent. However, a common topology is one that includes this hop, which is often another BIG-IP system.*

Alternate configuration

If the DHCP client subnet includes a BIG-IP system that serves as a hop to the BIG-IP relay agent, you must perform two additional configuration tasks:

- You must configure the BIG-IP relay agent to relay the client DHCP requests to the DHCP servers without losing the originating subnet (source) IP address. This originating source IP address is typically a self IP address of the BIG-IP system that resides on the client subnet. You configure the BIG-IP relay agent to preserve the originating source IP address by creating a SNAT that specifies the originating self IP address as both the origin address and the translation address. A SNAT configured in this way prevents the BIG-IP relay agent, before sending the DHCP broadcast message to the DHCP servers, from translating the source IP address of the incoming DHCP request to a different address.
- You must add a route (to the BIG-IP relay agent) that specifies the originating source IP address as the destination for DHCP responses. The DHCP servers use this route to send their responses back through the BIG-IP relay agent to the clients.

Task summary

You configure the BIG-IP system to act as a Dynamic Host Configuration Protocol (DHCP) relay agent by creating a pool of DHCP servers and then creating a virtual server to manage DHCP client broadcast messages.

Task list

Creating a pool of DHCP servers

Creating a DHCP Relay type virtual server

Creating a pool of DHCP servers

You must create a pool that includes Dynamic Host Configuration Protocol (DHCP) servers as pool members before you create a DHCP relay type virtual server.

Important: *Never add more than 20 pool members to a pool of DHCP servers. Although the BIG-IP[®] system allows you to add more than 20 members to a pool, only 20 members or fewer will receive the client DHCP DISCOVER request.*

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. (Optional) Type a description for the pool.
5. (Optional) For the **Health Monitors** setting, in the **Available** list, select **UDP**, and click << to move the monitor to the **Active** list.

6. From the **Load Balancing Method** list, select a method.

***Note:** A DHCP pool requires a load balancing method, although actual load balancing across DHCP pool members is ignored and DHCP requests are sent to all DHCP pool members.*

7. For the **Priority Group Activation** setting, select **Disabled**.
8. Add each resource that you want to include in the pool using the **New Members** setting:
 - a) (Optional) Type a name in the **Node Name** field, or select a node address from the **Node List**.
 - a) Type an IP address in the **Address** field, or select a node address from the **Node List**.
 - b) Type 67 (IPv4) or 547 (IPv6) in the **Service Port** field.
 - c) Click **Add**.
9. Click **Finished**.

A pool that includes DHCP servers as pool members is created.

Creating a DHCP Relay type virtual server

A DHCP relay type BIG-IP® virtual server provides you with the ability to relay DHCP client requests for an IP address to one or more DHCP servers, and provide DHCP server responses with an available IP address for the client.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. (Optional) Type a description for the virtual server.
5. From the **Type** list, select **DHCP Relay**.
6. Select one of the following to configure a **Destination** type.

Destination	Steps to configure
255.255.255.255 (IPv4 Default)	None.
ff02::1:2 (IPv6 Default)	None.
Other	Select and configure one of the following types. <ul style="list-style-type: none"> • Host, and type 255 . 255 . 255 . 255 in the Address field. • Network, type 255 . 255 . 255 . 255 in the Address field, and type 255 . 255 . 255 . 255 in the Mask field.
7. From the **State** list, select **Enabled**.
8. In the Configuration area for the **VLAN and Tunnel Traffic** setting, select the VLANs on the same network as the DHCP clients to ensure that the BIG-IP system can accept the broadcast traffic from the client.
9. From the **Default Pool** list, select the pool that is configured for DHCP servers.
10. Click **Finished**.

A DHCP relay type virtual server is configured to provide the ability to relay DHCP client requests for an IP address to one or more DHCP servers, and provide DHCP server responses with an available IP address for the client.

Implementation result

The BIG-IP[®] system is configured to manage Dynamic Host Configuration Protocol (DHCP) client IP addresses, using a DHCP Relay type virtual server to manage DHCP client broadcast messages.

Configuring the BIG-IP System for DHCP Renewal

Overview: Renewing IP addresses for DHCP clients

You can configure the BIG-IP[®] system to manage DHCP renewal requests and responses.

Before configuring the BIG-IP system to manage DHCP renewal requests and responses, it is helpful to understand some BIG-IP system terminology:

BIG-IP object type	Definition
BIG-IP pool member	A DHCP relay target (such as a DHCP server or BOOTP server). This is the dynamic address server to which the BIG-IP system forwards unicast requests.
BIG-IP virtual server	A BIG-IP system address on the listening VLAN
BIG-IP VLAN assigned to a virtual server	A listening VLAN, controlled on a per-virtual server basis

About DHCP renewal

You can configure the BIG-IP system to act as a DHCP renewal system. A common reason to configure the BIG-IP system as a renewal system is when the DHCP servers reside on a different subnet than that of the client systems, and the BIG-IP system is also configured as a DHCP relay agent. As a DHCP renewal system, the BIG-IP system manages the renewal of client IP addresses by DHCP servers before the addresses expire.

During the renewal process, a DHCP client sends a renewal request, which is passed through a BIG-IP Forwarding IP type of virtual server directly to the specific DHCP server that issued the initial client IP address. The DHCP server then sends a response to renew the lease for the client's IP address.

In the example shown in the illustration, a DHCP client sends a renewal message to the same BIG-IP system that initially acted as the DHCP relay agent. This renewal request is forwarded through a BIG-IP renewal virtual server directly to DHCP server 1. DHCP server 1 then provides a response to renew the lease for the client's IP address.

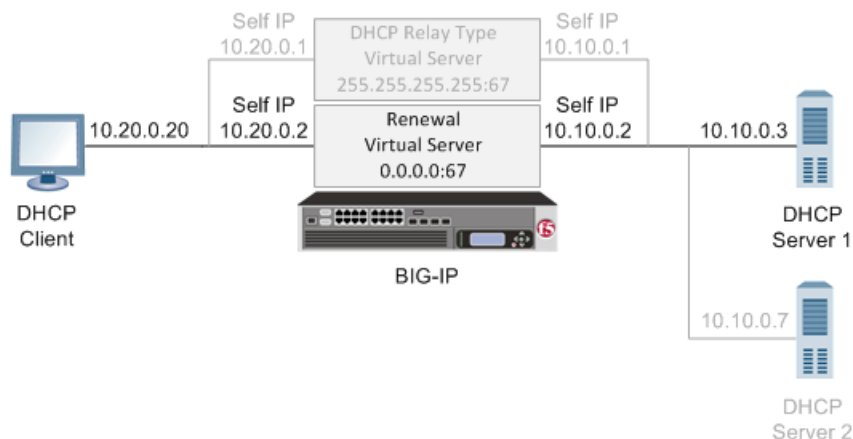


Figure 15: A sample DHCP renewal system configuration

Task summary

You configure a BIG-IP system to act as a Dynamic Host Configuration Protocol (DHCP) relay system by creating a virtual server that specifically forwards DHCP renewal requests to the appropriate DHCP server.

Task list

Creating a DHCP renewal virtual server

Creating a DHCP renewal virtual server

A Dynamic Host Configuration Protocol (DHCP) renewal virtual server forwards a DHCP request message from a DHCP client directly to a DHCP server, to automatically renew an IP address before it expires.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. (Optional) Type a description for the virtual server.
5. From the **Type** list, select **Forwarding (IP)**.
6. For a **Destination** type, select **Host**, and type the DHCP server IP address in the **Address** field.

***Tip:** If you have multiple DHCP servers, type 0.0.0.0 in the **Address** field.*

7. In the **Service Port** field, type 67 (IPv4) or 547 (IPv6).
8. From the **Protocol** list, select **UDP**.
9. From the **VLAN and Tunnel Traffic** list, select the VLANs on the same network as the DHCP clients.
10. Click **Finished**.

The BIG-IP system is now configured with a virtual server that can forward DHCP renewal requests directly to the appropriate DHCP server.

Implementation result

The BIG-IP® system is configured to forward DHCP client renewal requests to appropriate DHCP servers that reside on a different subnet than the client systems. The BIG-IP also forwards the DHCP server responses back to the client systems, therefore ensuring that client IP addresses do not expire.

Configuring a One-IP Network Topology

Overview: Configuring a one-IP network topology

One configuration option you can use with the BIG-IP® system is a one-IP network topology. This differs from the typical two-network configuration in two ways:

- Because there is only one physical network, this configuration does not require more than one interface on the BIG-IP system.
- Clients need to be assigned SNATs to allow them to make connections to servers on the network in a load balancing pool.

Part of this configuration requires you to configure the BIG-IP system to handle connections originating from the client. You must define a SNAT in order to change the source address on the packet to the SNAT external address, which is located on the BIG-IP system. Otherwise, if the source address of the returning packet is the IP address of the content server, the client does not recognize the packet because the client sent its packets to the IP address of the virtual server, not the content server.

If you do not define a SNAT, the server returns the packets directly to the client without giving the BIG-IP system the opportunity to translate the source address from the server address back to the virtual server. If this happens, the client might reject the packet as unrecognizable.

The single interface configuration is shown in the following illustration.

Illustration of a one-IP network topology for the BIG-IP system

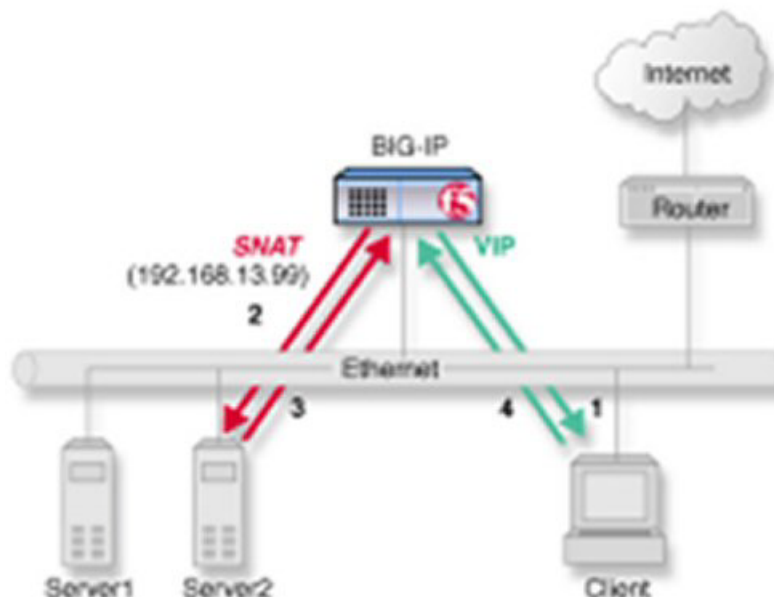


Figure 16: One-IP network topology for the BIG-IP system

Task summary for a one-IP network topology for the BIG-IP system

You can perform these tasks to configure a one-IP network topology.

Task list

Creating a pool for processing HTTP connections with SNATs enabled

Creating a virtual server for HTTP traffic

Defining a default route

Configuring a client SNAT

Creating a pool for processing HTTP connections with SNATs enabled

Verify that all content servers for the pool are in the network of VLAN **external**.

For a basic configuration, you need to create a pool to manage HTTP connections. This pool enables SNATs for any connections destined for a member of the pool.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, from the **Available** list, select the **http** monitor, and click << to move the monitor to the **Active** list.
5. For the **Allow SNAT** setting, verify that the value is **Yes**.
6. In the Resources area of the screen, use the default values for the **Load Balancing Method** and **Priority Group Activation** settings.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type 80 in the **Service Port** field, or select **HTTP** from the list.
 - c) (Optional) Type a priority number in the **Priority** field.
 - d) Click **Add**.
8. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for HTTP traffic

This task creates a destination IP address for application traffic. As part of this task, you must assign the relevant pool to the virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.

4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **HTTP Profile** list, select **http**.
7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. Click **Finished**.

You now have a virtual server to use as a destination address for application traffic.

Defining a default route

Another task that you must perform to implement one-IP network load balancing is to define a default route for the VLAN `external`.

1. On the Main tab, click **Network > Routes**.
2. Click **Add**.
The New Route screen opens.
3. In the **Name** field, type `Default Gateway Route`.
4. In the **Destination** field, type the IP address `0.0.0.0`.
An IP address of `0.0.0.0` in this field indicates that the destination is a default route.
5. From the **Resource** list, select **Use VLAN/Tunnel**.
A VLAN represents the VLAN through which the packets flow to reach the specified destination.
6. Select **external** from the **VLAN/Tunnel** list.
7. At the bottom of the screen, click **Finished**.

The default route for VLAN `external` is defined.

Configuring a client SNAT

To configure the BIG-IP® system to handle connections originating from the client, you can define a SNAT to change the source address on the packet to the SNAT external address located on the BIG-IP system.

1. On the Main tab, click **Local Traffic > Address Translation**.
The **SNAT List** screen displays a list of existing SNATs.
2. Click **Create**.
3. Name the new SNAT.
4. In the **Translation** field, type the IP address that you want to use as a translation IP address.
5. From the **Origin** list, select **Address List**.
6. For each client to which you want to assign a translation address, do the following:
 - a) Select **Host**.
 - b) Type a client IP address in the **Address** field.
 - c) Click **Add**.
7. From the **VLAN/Tunnel Traffic** list, select **Enabled on**.

8. For the **VLAN List** setting, in the **Available** field, select **external**, and using the **Move** button, move the VLAN name to the **Selected** field.
9. Click **Finished**.

The BIG-IP system is configured to handle connections originating from the client

Implementing Health and Performance Monitoring

Overview: Health and performance monitoring

You can set up the BIG-IP[®] system to monitor the health or performance of certain nodes or servers that are members of a load balancing pool. Monitors verify connections on pool members and nodes. A monitor can be either a health monitor or a performance monitor, designed to check the status of a pool, pool member, or node on an ongoing basis, at a set interval. If a pool member or node being checked does not respond within a specified timeout period, or the status of a pool member or node indicates that performance is degraded, the BIG-IP system can redirect the traffic to another pool member or node.

Some monitors are included as part of the BIG-IP system, while other monitors are user-created. Monitors that the BIG-IP system provides are called pre-configured monitors. User-created monitors are called custom monitors.

Before configuring and using monitors, it is helpful to understand some basic concepts regarding monitor types, monitor settings, and monitor implementation.

Monitor types

Every monitor, whether pre-configured or custom, is a certain type of monitor. Each type of monitor checks the status of a particular protocol, service, or application. For example, one type of monitor is HTTP. An HTTP type of monitor allows you to monitor the availability of the HTTP service on a pool, pool member, or node. A WMI type of monitor allows you to monitor the performance of a pool, pool member, or node that is running the Windows Management Instrumentation (WMI) software. An ICMP type of monitor simply determines whether the status of a node is up or down.

Monitor settings

Every monitor consists of settings with values. The settings and their values differ depending on the type of monitor. In some cases, the BIG-IP system assigns default values. For example, the following shows the settings and default values of an ICMP-type monitor.

```
Name my_icmp
Type ICMP
Interval 5
Timeout 16
Transparent No
Alias Address * All Addresses
```

Note: If you want to monitor the performance of a RealNetworks[®] RealServer server or a Windows[®]-based server equipped with Windows Management Instrumentation (WMI), you must first download a special plug-in file onto the BIG-IP system.

Task summary

To implement a health or performance monitor, you perform these tasks.

Task list

Creating a custom monitor

Creating a load balancing pool

Creating a virtual server

Creating a custom monitor

Before creating a custom monitor, you must decide on a monitor type.

You can create a custom monitor when the values defined in a pre-configured monitor do not meet your needs, or no pre-configured monitor exists for the type of monitor you are creating.

Important: When defining values for custom monitors, make sure you avoid using any values that are on the list of reserved keywords.

1. On the Main tab, click **Local Traffic > Monitors**.
The Monitor List screen opens.
2. Click **Create**.
The New Monitor screen opens.
3. Type a name for the monitor in the **Name** field.
4. From the **Type** list, select the type of monitor.
The screen refreshes, and displays the configuration options for the monitor type.
5. From the **Import Settings** list, select an existing monitor.
The new monitor inherits initial configuration values from the existing monitor.
6. From the Configuration list, select **Advanced**.
This selection makes it possible for you to modify additional default settings.
7. Configure all settings shown.
8. Click **Finished**.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: You must create the pool before you create the corresponding virtual server.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server

A virtual server represents a destination IP address for application traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. In the Resources area of the screen, from the **Default Pool** list, select a pool name.

The web customer now has a destination IP address on the BIG-IP system for application traffic.

Preventing TCP Connection Requests From Being Dropped

Overview: TCP request queuing

TCP request queuing provides the ability to queue connection requests that exceed the capacity of connections for a pool, pool member, or node, as determined by the connection limit. Consequently, instead of dropping connection requests that exceed the capacity of a pool, pool member, or node, TCP request queuing makes it possible for those connection requests to reside within a queue in accordance with defined conditions until capacity becomes available.

When using session persistence, a request becomes queued when the pool member connection limit is reached.

Without session persistence, when all pool members have a specified connection limit, a request becomes queued when the total number of connection limits for all pool members is reached.

Conditions for queuing connection requests include:

- The maximum number of connection requests within the queue, which equates to the maximum number of connections within the pool, pool member, or node. Specifically, the maximum number of connection requests within the queue cannot exceed the cumulative total number of connections for each pool member or node. Any connection requests that exceed the capacity of the request queue are dropped.
- The availability of server connections for reuse. When a server connection becomes available for reuse, the next available connection request in the queue becomes dequeued, thus allowing additional connection requests to be queued.
- The expiration rate of connection requests within the queue. As queue entries expire, they are removed from the queue, thus allowing additional connection requests to be queued.

Connection requests within the queue become dequeued when:

- The connection limit of the pool is increased.
- A pool member's slow ramp time limit permits a new connection to be made.
- The number of concurrent connections to the virtual server falls to less than the connection limit.
- The connection request within the queue expires.

Preventing TCP connection requests from being dropped

When you enable TCP request queuing, connection requests become queued when they exceed the total number of available server connections.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click a pool name in the Pool List.
3. From the **Configuration** list, select **Advanced**.
4. In the **Enable Request Queuing** list, select **Yes**.
5. In the **Request Queue Depth** field, type the maximum number of connections allowed in the queue.

***Note:** If you type zero (0) or leave the field blank, the maximum number of queued connections is unlimited, constrained only by available memory.*

6. In the **Request Queue Timeout** field, type the maximum number of milliseconds that a connection can remain queued.

***Note:** If you type zero (0) or leave the field blank, the maximum number of milliseconds is unlimited.*

7. Click **Update**.

Connection requests become queued when they exceed the total number of available server connections.

Setting Connection Limits

Overview: About connection limits

You can configure a virtual server, pool member, or node to prevent an excessive number of connection requests during events such as a Denial of Service (DoS) attack or a planned, high-demand traffic event. To ensure the availability of a virtual server, pool member, or node, you can use the BIG-IP® Local Traffic Manager™ to manage the total number of connections and the rate at which connections are made.

When you specify a connection limit, the system prevents the total number of concurrent connections to the virtual server, pool member, or node from exceeding the specified number.

When you specify a connection rate limit, the system controls the number of allowed new connections per second, thus providing a manageable increase in connections without compromising availability.

Limiting connections for a virtual server, pool member, or node

You can improve the availability of a virtual server, pool member, or node by using the BIG-IP® Local Traffic Manager™ to specify a connection limit and a connection rate limit.

1. On the Main tab, expand **Local Traffic**, and then click **Virtual Servers, Pools, or Nodes**.
2. Click the name of the virtual server, pool, or node you want to modify.
3. For virtual servers only, from the **Configuration** list, select **Advanced**.
4. In the **Connection Limit** field, type a number that specifies the maximum number of concurrent open connections.
5. In the **Connection Rate Limit** field, type a number that specifies the number of new connections accepted per second for the virtual server.
6. Click **Update** to save the changes.

After configuring connection and connection rate limits on a virtual server, or after configuring these limits on a pool member or node associated with a virtual server, the system controls the total number of concurrent connections and the rate of new connections to the virtual server, pool member, or node.

Implementation results

Configuring a connection limit or a connection rate limit for a virtual server, pool member, or node prevents an excessive number of connection requests during events such as a Denial of Service (DoS) attack or a planned, high-demand traffic event. In this way, you can manage the total number of connections to a virtual server, pool member, or node, as well as the rate at which connections are made. When you specify a connection rate limit, the system controls the number of allowed new connections per second, thus providing a manageable increase in connections without compromising availability.

Load Balancing to IPv6 Nodes

Overview: Load balancing to IPv6 nodes

To set up the BIG-IP® system to function as an IPv4-to-IPv6 gateway, you create a load balancing pool consisting of members that represent IPv6 nodes. You also create a virtual server that load balances traffic to those pool members.

As an option, you can use the `tmsh` command line interface to configure the BIG-IP system to send out ICMPv6 routing advisory messages, and to respond to ICMPv6 route solicitation messages. When you perform this task, the BIG-IP system begins to support auto-configuration of downstream nodes. Also, the downstream nodes automatically discover that the BIG-IP system is their router.

Task summary

When you configure IPv4-to-IPv6 load balancing, you must create a pool for load balancing traffic to IPv6 nodes, and then create an IPv4 virtual server that processes application traffic.

Task list

Creating a load balancing pool

Creating a virtual server for IPv6 nodes

Creating a load balancing pool

The first task in configuring IPv4-to-IPv6 load balancing is to create a pool to load balance connections to IPv6 nodes. Use the Configuration utility to create this pool.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.

- Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
 8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for IPv6 nodes

You can define a virtual server that references the pool of IPv6 nodes.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type a port number or select a service name from the **Service Port** list.
6. In the Resources area of the screen, from the **Default Pool** list, select the name of the pool that contains the IPv6 servers.
7. Click **Finished**.

The virtual server that references the pool of IPv6 nodes appears in the Virtual Servers list.

Mitigating Denial of Service Attacks

Overview: Mitigating Denial of Service and other attacks

The BIG-IP® system contains several features that provide you with the ability to create a configuration that contributes to the security of your network. In particular, the BIG-IP system is in a unique position to mitigate some types of Denial of Service (DoS) attacks that try to consume system resources in order to deny service to the intended recipients.

The following features of the BIG-IP system help it resist many types of DoS attacks:

- The BIG-IP kernel has a mechanism built in to protect against SYN Flood attacks by limiting simultaneous connections, and tearing down connections that have unacknowledged SYN/ACK packets after some time period as passed. (A SYN/ACK packet is a packet that is sent as part of the TCP three-way handshake).
- BIG-IP system can handle tens of thousands of Layer 4 (L4) connections per second. It would take a very determined attack to affect either the BIG-IP system itself, or the site, if sufficient server resources and bandwidth are available.
- SYN floods, or Denial of Service (DoS) attacks, can consume all available memory. The BIG-IP system supports a large amount of memory to help it resist DoS attacks.

Denial of Service attacks and iRules

You can create BIG-IP® iRules® to filter out malicious DoS attacks. After you identify a particular attack, you can write an iRule that discards packets containing the elements that identify the packet as malicious.

iRules for Code Red attacks

The BIG-IP® system is able to filter out the Code Red attack by using an iRule to send the HTTP request to a dummy pool.

```
when HTTP_REQUEST {  
    if {string tolower [HTTP::uri] contains "default.ida" } {  
        discard  
    } else {  
        pool RealServerPool  
    }  
}
```

iRules for Nimda attacks

The Nimda worm is designed to attack systems and applications based on the Microsoft® Windows® operating system.

```
when HTTP_REQUEST {
    set uri [string tolower [HTTP::uri]]
    if { ($uri contains "cmd.exe") or ($uri contains
        "root.exe") or ($uri contains "admin.dll") } {
        discard
    } else {
        pool ServerPool
    }
}
```

Common Denial of Service attacks

You might want to know how the BIG-IP® system reacts to certain common attacks that are designed to deny service by breaking the service or the network devices. The following information lists the most common attacks, along with how the BIG-IP system functionality handles the attack.

Attack type	Description	Mitigation
SYN flood	A <i>SYN flood</i> is an attack against a system for the purpose of exhausting that system's resources. An attacker launching a SYN flood against a target system attempts to occupy all available resources used to establish TCP connections by sending multiple SYN segments containing incorrect IP addresses. Note that the term SYN refers to a type of connection state that occurs during establishment of a TCP/IP connection. More specifically, a SYN flood is designed to fill up a SYN queue. A SYN queue is a set of connections stored in the connection table in the SYN-RECEIVED state, as part of the standard three-way TCP handshake. A SYN queue can hold a specified maximum number of connections in the SYN-RECEIVED state. Connections in the SYN-RECEIVED state are considered to be half-open and waiting for an acknowledgment from the client. When a SYN flood causes the maximum number of allowed connections in the SYN-RECEIVED state to be reached, the SYN queue is said to be full, thus preventing the target system from establishing other legitimate connections. A full SYN queue therefore results in partially-open TCP connections to IP addresses that either do not exist or are unreachable. In these cases, the connections must reach their timeout before the server can continue fulfilling other requests.	The BIG-IP system includes a feature designed to alleviate SYN flooding. Known as SYN Check™, this feature sends information about the flow, in the form of cookies, to the requesting client, so that the system does not need to keep the SYN-RECEIVED state that is normally stored in the connection table for the initiated session. Because the SYN-RECEIVED state is not kept for a connection, the SYN queue cannot be exhausted, and normal TCP communication can continue. The SYN Check feature complements the existing adaptive reaper feature in the BIG-IP system. While the adaptive reaper handles established connection flooding, SYN Check prevents connection flooding altogether. That is, while the adaptive reaper must work overtime to flush connections, the SYN Check feature prevents the SYN queue from becoming full, thus allowing the target system to continue to establish TCP connections.
ICMP flood (Smurf)	The <i>ICMP flood</i> , sometimes referred to as a Smurf attack, is an attack based on a method of making a remote network send ICMP Echo replies to a single host. In this attack, a single packet from the attacker goes to an unprotected network's broadcast address.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Attack type	Description	Mitigation
	Typically, this causes every machine on that network to answer with a packet sent to the target. The BIG-IP system is hardened against these attacks because it answers only a limited number of ICMP requests per second, and then drops the rest. On the network inside the BIG-IP system, the BIG-IP system ignores directed subnet broadcasts, and does not respond to the broadcast ICMP Echo that a Smurf attacker uses to initiate an attack.	
UDP flood	The <i>UDP flood</i> attack is most commonly a distributed Denial of Service attack (DDoS), where multiple remote systems are sending a large flood of UDP packets to the target. The BIG-IP system handles these attacks similarly to the way it handles a SYN flood. If the port is not listening, the BIG-IP system drops the packets. If the port is listening, the reaper removes the false connections.	Setting the UDP idle session timeout to between 5 and 10 seconds reaps these connections quickly without impacting users with slow connections. However, with UDP this might still leave too many open connections, and your situation might require a setting of between 2 and 5 seconds.
UDP fragment	The <i>UDP fragment</i> attack is based on forcing the system to reassemble huge amounts of UDP data sent as fragmented packets. The goal of this attack is to consume system resources to the point where the system fails. The BIG-IP system does not reassemble these packets, it sends them on to the server if they are for an open UDP service. If these packets are sent with the initial packet opening the connection correctly, then the connection is sent to the back-end server. If the initial packet is not the first packet of the stream, the entire stream is dropped.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.
Ping of Death	The <i>Ping of Death</i> attack is an attack with ICMP echo packets that are larger than 65535 bytes. As this is the maximum allowed ICMP packet size, this can crash systems that attempt to reassemble the packet. The BIG-IP system is hardened against this type of attack. However, if the attack is against a virtual server with the Any IP feature enabled, then these packets are sent on to the server. It is important that you apply the latest updates to your servers.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.
Land	A <i>Land</i> attack is a SYN packet sent with the source address and port the same as the destination address and port. The BIG-IP system is hardened to resist this attack. The BIG-IP system connection table matches existing connections so that a spoof of this sort is not passed on to the servers. Connections to the BIG-IP system are checked and dropped if spoofed in this manner.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.
Teardrop	A <i>Teardrop</i> attack is carried out by a program that sends IP fragments to a machine connected to the Internet or a network. The Teardrop attack exploits an overlapping IP fragment problem present in some common operating systems. The problem causes the TCP/IP fragmentation re-assembly code to improperly handle overlapping IP fragments. The BIG-IP system handles these attacks by correctly checking frame alignment and discarding improperly aligned fragments.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.
Data	The BIG-IP system can also offer protection from data attacks to the servers behind the BIG-IP system. The BIG-IP system acts as a port-deny device, preventing many common exploits by simply not passing the attack through to the server.	You do not need to make any changes to the BIG-IP system configuration for this type of attack.

Attack type	Description	Mitigation
WinNuke	The <i>WinNuke</i> attack exploits the way certain common operating systems handle data sent to the NetBIOS ports. NetBIOS ports are 135, 136, 137 and 138, using TCP or UDP. The BIG-IP system denies these ports by default.	On the BIG-IP system, do not open these ports unless you are sure your servers have been updated against this attack.
Sub 7	The <i>Sub 7</i> attack is a Trojan horse that is designed to run on certain common operating systems. This Trojan horse makes it possible the system to be controlled remotely. This Trojan horse listens on port 27374 by default. The BIG-IP system does not allow connections to this port from the outside, so a compromised server cannot be controlled remotely.	Do not open high ports (ports higher than 1024) without explicit knowledge of what applications will be running on these ports.
Back Orifice	A <i>Back Orifice</i> attack is a Trojan horse that is designed to run on certain common operating systems. This Trojan horse makes it possible the system to be controlled remotely. This Trojan horse listens on UDP port 31337 by default. The BIG-IP system does not allow connections to this port from the outside, so a compromised server cannot be controlled remotely.	Do not open high ports (ports higher than 1024) without explicit knowledge of what will be running on these ports

Task summary

There are several tasks you can perform to mitigate Denial of Service attacks.

Task list

[Configuring adaptive reaping](#)
[Setting the TCP and UDP connection timers](#)
[Applying a rate class to a virtual server](#)
[Calculating connection limits on the main virtual server](#)
[Setting connection limits on the main virtual server](#)
[Adjusting the SYN Check threshold](#)

Configuring adaptive reaping

This procedure configures adaptive reaping. The *adaptive connection reaper* closes idle connections when memory usage on the BIG-IP system increases. This feature makes it possible for the BIG-IP system to aggressively reap connections when the system memory utilization reaches the low-water mark, and to stop establishing new connections when the system memory utilization reaches the high-water mark percentage.

If the BIG-IP platform includes an LCD panel, an adaptive reaping event causes the BIG-IP system to display the following message on the LCD panel:

Blocking DoS attack

Warning: The adaptive reaper settings do not apply to SSL connections. However, you can set TCP and UDP connection timeouts that reap idle SSL connections.

1. On the Main tab, click **System > Configuration**.
The General screen opens.

2. From the Local Traffic menu, choose **General**.
3. In the Properties area of the screen, set the **Reaper High-water Mark** property to 95.
4. Set the **Reaper Low-water Mark** property to 85.
5. Click **Update**.

When aggressive mode is activated on the BIG-IP system, the event is marked in the `/var/log/ltm` file with messages similar to these examples:

```
tmm tmm[PID]: 011e0002:4: sweeper_update: aggressive mode activated. (117504/138240 pages)
```

```
tmm tmm[PID]: 011e0002:4: sweeper_update: aggressive mode deactivated. (117503/138240 pages)
```

Important: *Setting both of the adaptive reaper values to 100 disables this feature.*

Setting the TCP and UDP connection timers

You can set the TCP and UDP timers in the profile settings for the TCP profile and the UDP profiles. You should set these timers for the services that you use for your virtual servers. For example, you can set a value of 60 for HTTP connections and 60 for SSL connections.

1. On the Main tab, click **Local Traffic > Profiles**.
2. From the **Protocol** menu, choose TCP or UDP.
3. Click the name of the profile type you want to configure.
4. Set the **Idle Timeout** setting to 60.
5. Click **Update**.

Applying a rate class to a virtual server

After you create a rate class, you can apply it to the virtual servers in the configuration.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. In the **Virtual Server** list, click the virtual server that you want.
3. In the **Configuration** list, click **Advanced**.
4. In the **Rate Class** list, select a rate class.
5. Click **Update**.

The rate class is applied to the virtual server.

Calculating connection limits on the main virtual server

Use this procedure to determine a connection limit.

Before you set a connection limit, use the following formula to calculate the connection limit value for the main virtual server:

$$\text{Connection Limit} = \text{Approximate Amount of RAM in KB} * 0.8.$$

For example, if you have 256 MB of RAM, the calculation is:

$$256,000 * 0.8 = 204800$$

In this case, you set the connection limit to **204800**.

Setting connection limits on the main virtual server

Connection limits determine the maximum number of concurrent connections allowed on a virtual server. In this context, the main virtual server is the virtual server that receives the most traffic to your site.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the virtual server that you want to modify.
3. From the **Configuration** list, select **Advanced**.
4. In the **Connection Limit** field, type the number that you calculated for the connection limit.
5. Click **Update** to save the changes.

The virtual server is configured for the specified maximum number of concurrent connections.

Adjusting the SYN Check threshold

You can configure the SYN Check™ feature to prevent the BIG-IP SYN queue from becoming full during a SYN flood attack. The **SYN Check Activation Threshold** setting indicates the number of new or untrusted TCP connections that can be established before the BIG-IP activates the SYN Cookies authentication method for subsequent TCP connections.

1. On the Main tab, click **System > Configuration**.
2. From the Local Traffic menu, choose General.
3. In the **SYN Check Activation Threshold** field, type the number of connections that you want to define for the threshold.
4. Click **Update**.

If SYN flooding occurs, the BIG-IP system now protects the BIG-IP SYN queue from becoming full.

Configuring Remote CRLDP Authentication

Overview of remote authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task Summary

To configure remote authentication with CRLDP, you must create a configuration object and a profile that correspond to the authentication server you are using to store your user accounts. You must also create a third type of object. This object is referred to as a server object.

Task list

Creating a CRLDP configuration object for authenticating application traffic remotely

Creating a custom CRLDP profile

Modifying a virtual server for CRLDP authentication

Creating a CRLDP configuration object for authenticating application traffic remotely

The CRLDP authentication module verifies the revocation status of an SSL certificate, as part of authenticating that certificate. A *CRLDP configuration object* specifies information that the BIG-IP system needs to perform the remote authentication.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.

4. In the **Name** field, type a unique name for the configuration object, such as `asmy_crl dp_config`.
5. From the **Type** list, select **CRLDP**.
6. In the **Connection Timeout** field, retain or change the time limit, in seconds, for the connection to the Certificate Revocation List Distribution Points (CRLDP) server.
7. In the **Update Interval** field, retain or change the interval, in seconds, for the system to use when receiving updates from the CRLDP server.
If you use the default value of 0 (zero), the CRLDP server updates the system according to the expiration time specified for the CRL.
8. For the **Use Issuer** setting, retain the default value (cleared) or select the box.
When cleared (disabled), the BIG-IP system extracts the CRL distribution point from the incoming client certificate. When selected (enabled), the BIG-IP system extracts the CRL distribution point from the signing certificate.
9. For the **CRLDP Servers** setting, select a CRLDP server name in the **Available** list, and using the Move button, move the name to the **Selected** list.
10. Click **Finished**.

You now have a CRLDP configuration object that a CRLDP profile can reference.

Creating a custom CRLDP profile

The next task in configuring CRLDP-based remote authentication on the BIG-IP® system is to create a custom CRLDP profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **CRLDP** from the **Type** list.
5. Select `ssl_crl dp` in the **Parent Profile** list.
6. Select the **Custom** check box.
7. Select a CRLDP configuration object from the **Configuration** list.
8. Click **Finished**.

Modifying a virtual server for CRLDP authentication

The final task in the process of implementing CRLDP authentication is to assign the custom CRLDP profile to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a virtual server.
3. From the **Configuration** list, select **Advanced**.
4. For the **Authentication Profiles** setting, in the **Available** field, select a custom CRLDP profile, and using the **Move** button, move the custom CRLDP profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom CRLDP profile.

Configuring Remote LDAP Authentication

Overview of remote LDAP authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task Summary

To configure remote authentication for LDAP traffic, you must create a configuration object and a profile that correspond to the LDAP authentication server you are using to store your user accounts. You must also modify the relevant virtual server.

Task list

Creating an LDAP configuration object for authenticating application traffic remotely

Creating a custom LDAP profile

Modifying a virtual server for LDAP authentication

Creating an LDAP configuration object for authenticating application traffic remotely

An *LDAP configuration object* specifies information that the BIG-IP system needs to perform the remote authentication. For example, the configuration object specifies the remote LDAP tree that the system uses as the source location for the authentication data.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.

4. In the **Name** field, type a unique name for the configuration object, such as `asmy_ldap_config`.
5. From the **Type** list, select **LDAP**.
6. In the **Remote LDAP Tree field**, type the file location (tree) of the user authentication database on the LDAP or Active Directory server.
At a minimum, you must specify a domain component (that is, **dc=value**).
7. In the **Hosts** field, type the IP address of the remote LDAP or Active Directory server.
8. Click **Add**.
The IP address of the remote LDAP or Active Directory server appears in the **Hosts** area.
9. Retain or change the **Service Port** value.
10. Retain or change the **LDAP Version** value.
11. Click **Finished**.

You now have an LDAP configuration object that the LDAP authentication profile can reference.

Creating a custom LDAP profile

The next task in configuring LDAP-based or Active Directory-based remote authentication on the BIG-IP® system is to create a custom LDAP profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **LDAP** from the **Type** list.
5. Select **ldap** in the **Parent Profile** list.
6. Select the LDAP configuration object that you created from the **Configuration** list.
7. Click **Finished**.

The custom LDAP profile appears in the **Profiles** list.

Modifying a virtual server for LDAP authentication

The final task in the process of implementing authentication using a remote LDAP server is to assign the custom LDAP profile and a default LDAP authentication iRule to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a Standard type of virtual server to which an HTTP profile is assigned.
3. From the **Configuration** list, select **Advanced**.
4. For the **Authentication Profiles** setting, in the **Available** field, select a custom LDAP profile, and using the **Move** button, move the custom LDAP profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom LDAP profile.

Configuring Remote RADIUS Authentication

Overview of remote authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP® system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task summary for RADIUS authentication of application traffic

To configure remote authentication for RADIUS traffic, you must create a configuration object and a profile that correspond to the RADIUS authentication server you are using to store your user accounts. You must also create a third type of object. This object is referred to as a server object.

Task list

Creating a RADIUS server object for authenticating application traffic remotely

Creating a RADIUS configuration object for authenticating application traffic remotely

Creating a custom RADIUS profile

Modifying a virtual server for RADIUS authentication

Creating a RADIUS server object for authenticating application traffic remotely

A *RADIUS server object* represents the remote RADIUS server that the BIG-IP system uses to access authentication data.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **RADIUS Servers**.
3. Click **Create**.

4. In the **Name** field, type a unique name for the server object, such as `asmy_radius_server`.
5. In the **Host** field, type the host name or IP address of the RADIUS server.
6. In the **Service Port** field, type the port number for RADIUS authentication traffic, or retain the default value (1812).
7. In the **Secret** field, type the secret key used to encrypt and decrypt packets sent or received from the server.
8. In the **Confirm Secret** field, re-type the secret you specified in the **Secret** field.
9. In the **Timeout** field, type a timeout value, in seconds, or retain the default value (3).
10. Click **Finished**.

You now have a RADIUS server object that the RADIUS configuration object can reference.

Creating a RADIUS configuration object for authenticating application traffic remotely

The BIG-IP system configuration must include at least one RADIUS server object.

You use a RADIUS authentication module when your authentication data is stored on a remote RADIUS server. A *RADIUS configuration object* specifies information that the BIG-IP system needs to perform the remote authentication.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.
4. In the **Name** field, type a unique name for the configuration object, such as `asmy_radius_config`.
5. From the **Type** list, select **RADIUS**.
6. For the **RADIUS Servers** setting, select a RADIUS server name in the **Available** list, and using the Move button, move the name to the **Selected** list.
7. In the **Client ID** field, type a string for the system to send in the **Network Access Server (NAS)-Identifier** RADIUS attribute.
8. Click **Finished**.

You now have a RADIUS configuration object that a RADIUS profile can reference.

Creating a custom RADIUS profile

The next task in configuring RADIUS-based remote authentication on the BIG-IP[®] system is to create a custom RADIUS profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **RADIUS** from the **Type** list.
5. Select **radius** in the **Parent Profile** list.
6. Select the RADIUS configuration object that you created from the **Configuration** list.
7. Click **Finished**.

The custom RADIUS profile appears in the **Profiles** list.

Modifying a virtual server for RADIUS authentication

The final task in the process of implementing authentication using a remote RADIUS server is to assign the custom RADIUS profile to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a virtual server.
3. From the **Configuration** list, select **Advanced**.
4. For the **Authentication Profiles** setting, in the **Available** field, select a custom RADIUS profile, and using the **Move** button, move the custom RADIUS profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom RADIUS profile.

Configuring Remote SSL LDAP Authentication

Overview of remote SSL LDAP authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task Summary

To configure remote authentication for SSL LDAP traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts.

Task list

Creating an LDAP Client Certificate SSL configuration object

Creating a custom SSL Client Certificate LDAP profile

Modifying a virtual server for SSL Client Certificate LDAP authorization

Creating an LDAP Client Certificate SSL configuration object

An *SSL Client Certificate LDAP configuration object* specifies information that the BIG-IP system needs to perform the remote authentication. This configuration object is one of the required objects you need to impose certificate-based access control on application traffic.

1. On the Main tab of the navigation pane, click **Local Traffic** > **Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.
4. In the **Name** field, type a unique name for the configuration object, such as `asmy_ssl_ldap_config`.

5. From the **Type** list, select **SSL Client Certificate LDAP**.
6. In the **Hosts** field, type an IP address for the remote LDAP authentication server storing the authentication data, and click **Add**.
The IP address appears in the **Hosts** area of the screen.
7. Repeat the previous step for each LDAP server you want to use.
8. From the **Search Type** list, select one of the following:

Option	Description
User	Choose this option if you want the system to extract a user name from the client certificate and search for that user name in the remote LDAP database.
Certificate Map	Choose this option if you want the system to search for an existing user-certificate mapping in the remote LDAP database.
Certificate	Choose this option if you want the system to search for a certificate stored in the user's profile in the remote LDAP database.

9. Click **Finished**.

You now have a configuration object that an SSL Client Certificate LDAP profile can reference.

Creating a custom SSL Client Certificate LDAP profile

The next task in configuring LDAP-based remote authentication on the BIG-IP[®] system is to create a custom SSL Client Certificate LDAP profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. Select **SSL Client Certificate LDAP** from the **Type** list.
6. Select **ssl_cc_ldap** in the **Parent Profile** list.
7. Select the name of a LDAP configuration object from the **Configuration** list.
8. Click **Finished**.

The custom SSL Client Certificate LDAP profile appears in the **Profiles** list.

Modifying a virtual server for SSL Client Certificate LDAP authorization

The final task in the process of implementing authorization using a remote LDAP server is to assign the custom SSL Client Certificate LDAP profile and a default LDAP authentication iRule to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a Standard-type virtual server to which an HTTP server profile is assigned.
3. From the **Configuration** list, select **Advanced**.

4. For the **Authentication Profiles** setting, in the **Available** field, select a custom SSL Client Certificate LDAP profile, and using the **Move** button, move the custom SSL Client Certificate LDAP profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom SSL Client Certificate LDAP profile.

Configuring Remote SSL OSCP Authentication

Overview of remote authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OSCP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OSCP authentication, this object is referred to as an OSCP responder.

Task Summary

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts.

When implementing an SSL OSCP authentication module, you must also create a third type of object. This object is referred to as an OSCP responder.

Task list

Creating an SSL OSCP responder object for authenticating application traffic remotely

Creating an SSL OSCP configuration object for authenticating application traffic remotely

Creating a custom SSL OSCP profile

Modifying a virtual server for SSL OSCP authentication

Creating an SSL OSCP responder object for authenticating application traffic remotely

An *SSL OSCP responder object* is an object that you create that includes a URL for an external SSL OSCP responder. You must create a separate SSL OSCP responder object for each external SSL OSCP responder.

1. On the Main tab of the navigation pane, click **Local Traffic** > **Profiles**.
2. From the Authentication menu, choose **OCSP Responders**.

3. Click **Create**.
4. In the **Name** field, type a unique name for the responder object, such `asmy_ocsp_responder`.
5. In the **URL** field, type the URL that you want the BIG-IP system to use to contact the Online Certificate Status Protocol (OCSP) service on the responder.
6. In the **Certificate Authority File** field, type the name of the file containing trusted Certificate Authority (CA) certificates that the BIG-IP system uses to verify the signature on the OCSP response.

You now have a responder that the SSL OCSP configuration object can reference.

Creating an SSL OCSP configuration object for authenticating application traffic remotely

The BIG-IP system configuration must include at least one SSL OCSP responder object.

An *SSL OCSP authentication module* checks the revocation status of an SSL certificate during remote authentication, as part of authenticating that certificate.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.
4. In the **Name** field, type a unique name for the configuration object, such `asmy_ocsp_config`.
5. From the **Type** list, select **SSL OCSP**.
6. For the **Responders** setting, select a responder server name from the **Available** list, and using the Move button, move the name to the **Selected** list.
7. Click **Finished**.

You now have an SSL OCSP configuration object that an SSL OCSP profile can reference.

Creating a custom SSL OCSP profile

The next task in configuring SSL OCSP-based remote authentication on the BIG-IP[®] system is to create a custom SSL OCSP profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **SSL OCSP** from the **Type** list.
5. Select the **Custom** check box.
6. Select an SSL OCSP configuration object from the **Configuration** list.
7. Select `ssl_ocsp` in the **Parent Profile** list.
8. Click **Finished**.

The custom SSL OCSP profile appears in the **Profiles:Authentication:Profiles** list.

Modifying a virtual server for SSL OSCP authentication

The final task in the process of implementing SSL OSCP authentication is to assign the custom SSL OSCP profile to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a virtual server.
3. From the **Configuration** list, select **Advanced**.
4. For the **Authentication Profiles** setting, in the **Available** field, select a custom SSL OSCP profile, and using the **Move** button, move the custom SSL OSCP profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom SSL OSCP profile.

Configuring Remote TACACS+ Authentication

Overview of remote authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP[®] system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task Summary

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts.

Task list

Creating a TACACS+ configuration object

Creating a custom TACACS+ profile

Modifying a virtual server for TACACS+ authentication

Creating a TACACS+ configuration object

A *TACACS+ configuration object* specifies information that the BIG-IP system needs to perform the remote authentication. For example, the configuration object specifies the IP address of the remote TACACS+ server.

1. On the Main tab of the navigation pane, click **Local Traffic** > **Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.
4. In the **Name** field, type a unique name for the configuration object, such as `asm_y_tacacs_config`.

5. From the **Type** list, select **TACACS+**.
6. For the **Servers** setting, select a server name in the **Available** list, and using the Move button, move the name to the **Selected** list.
7. In the **Secret** field, type the secret key used to encrypt and decrypt packets sent or received from the server.
Do not use the pound sign (#) in the secret for TACACS+ servers.
8. In the **Confirm Secret** field, re-type the secret you specified in the **Secret** field.
9. From the **Encryption** list, select an encryption option:

Option	Description
Enabled	Choose this option if you want the system to encrypt the TACACS+ packets.
Disabled	Choose this option if you want the system to send unencrypted TACACS+ packets.
10. In the **Service Name** field, type the name of the service that the user is requesting to be authenticated for use; typically, `ppp`.
Specifying the service makes it possible for the TACACS+ server to behave differently for different types of authentication requests. Examples of service names that you can specify are: `ppp`, `slip`, `arap`, `shell`, `tty-daemon`, `connection`, `system`, and `firewall`.
11. In the **Protocol Name** field, type the name of the protocol associated with the value specified in the **Service Name** field.
This value is usually `ip`. Examples of protocol names that you can specify are: `ip`, `lcp`, `ipx`, `stalk`, `vines`, `lat`, `xremote`, `tn3270`, `telnet`, `rlogin`, `pad`, `vpdn`, `ftp`, `http`, `deccp`, `osicp`, and `unknown`.
12. Click **Finished**.

You now have a configuration object that a TACACS+ authentication profile can reference.

Creating a custom TACACS+ profile

The next task in configuring TACACS+-based remote authentication on the BIG-IP® system is to create a custom TACACS+ profile.

1. On the Main tab, click **Local Traffic > Profiles > Authentication > Profiles**.
The Profiles list screen opens.
2. Click **Create**.
The New Authentication Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select **TACACS+** from the **Type** list.
5. Select **tacacs** in the **Parent Profile** list.
6. Select the TACACS+ configuration object that you created from the **Configuration** list.
7. Click **Finished**.

The custom TACACS+ profile appears in the **Profiles** list.

Modifying a virtual server for TACACS+ authentication

The final task in the process of implementing authentication using a remote TACACS+ server is to assign the custom TACACS+ profile and an existing default authentication iRule to a virtual server that is configured to process HTTP traffic (that is, a virtual server to which an HTTP profile is assigned).

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of a virtual server.
3. From the **Configuration** list, select **Advanced**.
4. For the **Authentication Profiles** setting, in the **Available** field, select a custom TACACS+ profile, and using the **Move** button, move the custom TACACS+ profile to the **Selected** field.
5. Click **Update** to save the changes.

The virtual server is assigned the custom TACACS+ profile.

Configuring Kerberos Delegation

Overview of remote authentication for application traffic

As an administrator in a large computing environment, you can set up the BIG-IP® system to use this server to authenticate any network traffic passing through the BIG-IP system. This type of traffic passes through a virtual server and through Traffic Management Microkernel (TMM) interfaces. Remote authentication servers typically use one of these protocols:

- Lightweight Directory Access Protocol (LDAP)
- Remote Authentication Dial-in User Service (RADIUS)
- TACACS+ (derived from Terminal Access Controller Access Control System [TACACS])
- Online Status Certificate Protocol (OCSP)
- Certificate Revocation List Distribution Point (CRLDP)
- Kerberos

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts. For example, if your remote authentication server is an LDAP server, you create an LDAP configuration object and an LDAP profile. When implementing a RADIUS, SSL OCSP, or CRLDP authentication module, you must also create a third type of object. For RADIUS and CRLDP authentication, this object is referred to as a server object. For SSL OCSP authentication, this object is referred to as an OCSP responder.

Task Summary

To configure remote authentication for this type of traffic, you must create a configuration object and a profile that correspond to the type of authentication server you are using to store your user accounts.

Task list

Creating a Kerberos Delegation configuration object

Creating a Kerberos delegation profile object from the command line

Creating a load balancing pool

Creating a virtual server with Kerberos delegation and Client SSL profiles

Creating a Kerberos Delegation configuration object

Use this procedure to create a configuration object for Kerberos delegation.

1. On the Main tab of the navigation pane, click **Local Traffic > Profiles**.
2. From the Authentication menu, choose **Configurations**.
3. Click **Create**.
4. In the **Name** field, type a unique name for the configuration object, such as `asmy_kerberos_config`.
5. From the **Type** list, select **Kerberos Delegation**.

6. For the **Enable Protocol Transition** setting, retain the default value (cleared) or select the box.
7. In the **Client Principal Name** field, type the name of the client principal, using the format HTTP/[name], where name is the name of the virtual server you created to use here.
This principal might be in a different domain from the server principal. If so, you should use the `domaintool (1)` utility to create this principal, because the client principal must have the **OK to Delegate** flag selected in the Microsoft Windows domain.
8. In the **Server Principal Name** field, type the name of the server principal (the back-end web server), using the format HTTP/[fqdn], where fqdn is the fully-qualified domain name.
This principal might be in a different domain from the client principal. If so, you should use the `domaintool (1)` utility to add the domain. Also, you probably need to use the `--dnsdomain` option to set up DNS-to-Kerberos realm mappings.
9. Click **Finished**.

Creating a Kerberos delegation profile object from the command line

You can create the Kerberos delegation profile object from the command line.

Set a cookie name and strong password for the cookie encryption key on the profile.

In this example, the cookie name is `kerbc` and the key is `kerbc: create profile auth my_kerberos_profile { configuration my_kerberos_config cookie-name kerbc cookie-key kerbc defaults-from krbdelegate }`

***Note:** The Cookie Key value is an encryption key that encrypts cookie data. A default value is supplied; however, you should change the default value so that attackers who know this value cannot decrypt cookie data and impersonate trusted users.*

The Kerberos delegation profile object is available.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

***Note:** You must create the pool before you create the corresponding virtual server.*

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

***Tip:** Hold the Shift or Ctrl key to select more than one monitor at a time.*

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.

The default is **Round Robin**.

6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server with Kerberos delegation and Client SSL profiles

You can create a virtual server with Kerberos delegation and Client SSL profiles.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type 80, or select **HTTP** from the list.
6. From the **Configuration** list, select **Advanced**.
7. From the **Type** list, select **Standard**.
8. From the **Protocol** list, select **TCP**.
9. From the **HTTP Profile** list, select **http**.
10. From the **SSL Profile (Client)** list, select a custom Client SSL profile.
11. For the **Authentication Profiles** setting, in the **Available** field, select a custom Kerberos delegation, and using the **Move** button, move the custom Kerberos delegation to the **Selected** field.
12. From the **Default Pool** list, select a pool name.
13. Click **Finished**.

The virtual server with Kerberos delegation and Client SSL profiles appears in the Virtual Server list.

Load Balancing Diameter Application Requests

Overview: Diameter load balancing

An optional feature of the BIG-IP® system is its ability to load balance and persist requests that applications send to servers running Diameter services. The BIG-IP system can also monitor each server to ensure that the Diameter service remains up and running.

Task summary

You implement Diameter load balancing by creating various local traffic objects in an administrative partition.

Task list

Creating a custom Diameter profile

Creating a custom Diameter monitor

Creating a pool to manage Diameter traffic

Creating a virtual server to manage Diameter traffic

Creating a custom Diameter profile

The first task in configuring Diameter load balancing on the BIG-IP® system is to create a custom Diameter profile.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **Diameter**.
The Diameter profile list screen opens.
2. Click **Create**.
The New Diameter profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Click **Finished**.

The custom Diameter profile appears in the **New Diameter Profile** list.

Creating a custom Diameter monitor

After you create a Diameter profile, you can create a custom Diameter monitor. The purpose of the Diameter monitor is to monitor the health of all servers running the Diameter service.

1. On the Main tab, click **Local Traffic** > **Monitors**.
2. Click **Create**.

3. In the **Name** field, type a unique name for the monitor, such as **my_diameter_monitor**.
4. From the **Type** list, select **Diameter**.
5. Retain the default values for all other settings.
6. Click **Finished**.

Creating a pool to manage Diameter traffic

The next step in a basic Diameter load balancing configuration is to define a load balancing pool that contains Diameter servers as its members.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
6. Click **Finished**.

The pool is configured to manage Diameter servers as pool members.

Creating a virtual server to manage Diameter traffic

The final task in configuring Diameter load balancing is to define a virtual server that references the custom Diameter profile and Diameter pool that you created in previous tasks.

Note: The virtual server to which you assign the Diameter profile must be a Standard type of virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. From the **Configuration** list, select **Advanced**.
6. From the **Diameter Profile** list, select a profile.

7. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
8. Click **Finished**.

The virtual server that references the custom Diameter profile and Diameter pool appears in the Virtual Server list.

Configuring the BIG-IP System for Electronic Trading

Overview: Configuring the BIG-IP system for electronic trading

The BIG-IP® system Local Traffic Manager™ (LTM®) FIX profile provides you with the ability to use Financial Information eXchange (FIX) protocol messages in routing, load balancing, persisting, and logging connections. The BIG-IP system uses the FIX profile to examine the header, body, and footer of each FIX message, and then process each message according to the parameters that it contains.

The BIG-IP system supports FIX protocol versions 4.2, 4.4, and 5.0, and uses the key-value pair FIX message format.

Important: *You cannot configure or use the BIG-IP FIX Profile to provide low-latency electronic trading functionality. Instead, you must implement low-latency electronic trading functionality separately. Refer to Implementing Low-Latency Electronic Trading Functionality for details.*

Task summary

There are several tasks you can perform to implement electronic trading.

Task list

Creating a data group list for a FIX profile

Creating a FIX profile for electronic trading

Creating a load balancing pool

Creating a virtual server for secure electronic trading

Viewing FIX message statistics

Creating a data group list for a FIX profile

You can create a data group list for a FIX profile that enables you to provide tag substitution, as required.

1. On the Main tab, click **Local Traffic** > **iRules** > **Data Group List**.
The Data Group List screen opens, displaying a list of data groups on the system.
2. Click **Create**.
The New Data Group screen opens.
3. In the **Name** field, type a unique name for the data group.
4. From the **Type** list, select **Integer**.
5. Using the **Integer Records** setting, create tag mapping entries consisting of an integer (client tag) and a value (server tag):
 - a) In the **Integer** field, type a value to be used for a specific client.
 - b) In the **Value** field, type a value that is substituted on the server.

- c) Click **Add**.

The new mapping between the integer and corresponding value appears in the list of Integer Records.

6. Click **Finished**.

The new data group appears in the list of data groups.

A data group list for a FIX profile is available.

Creating a FIX profile for electronic trading

You can create a FIX profile for electronic trading, and steer traffic in accordance with specified parameters.

1. On the Main tab, click **Local Traffic > Profiles > Services > FIX**.

The FIX profile list screen opens.

2. Click **Create**.

The New FIX Profile screen opens.

3. In the **Name** field, type a unique name for the profile.

4. From the **Parent Profile** list, select a parent profile.

5. Select the **Custom** check box.

6. (Optional) From the **Report Log Publisher** list, select the publisher for error messages and status reports.

7. (Optional) From the **Message Log Publisher** list, select the publisher for message logging.

8. In the **Rate Sample Interval** field, type the sample interval, in seconds, for the message rate.

9. From the **Error Action** list, select one of the following settings.

- **Don't Forward** (default) to drop a message with errors and not forward it.
- **Drop Connection** to disconnect the connection.

10. Select the **Quick Parsing** check box to parse the basic standard fields, and validate the message length and checksum.

11. Select the **Response Parsing** check box to parse the messages from the FIX server, applying the same parser configuration and error handling for the server as for the client.

12. Select the **Fully Parse Logon Message** check box to fully parse the logon message, instead of using quick parsing.

13. From the **Sender and Tag Substitution Data Group Mapping** list, select one of the following settings.

Setting	Description
Not Configured (default)	Disables the tag substitution map between sender ID and tag substitution data group.
Specify	Provides the Mapping List settings for you to configure as required. <ol style="list-style-type: none">1. In the Sender field, type a sender ID that represents the identity of the firm sending the message. Example: <code>client1</code>2. In the Data Group field, type a tag substitution data group. Example: <code>FIX_tag_map</code>3. Click Add.

14. Click **Finished**.

The FIX profile is configured for electronic trading.

Creating a load balancing pool

You can create a *load balancing pool* (a logical set of devices such as web servers that you group together to receive and process traffic) to efficiently distribute the load on your server resources.

Note: You must create the pool before you create the corresponding virtual server.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. For the **Health Monitors** setting, in the **Available** list, select a monitor type, and click << to move the monitor to the **Active** list.

Tip: Hold the Shift or Ctrl key to select more than one monitor at a time.

5. From the **Load Balancing Method** list, select how the system distributes traffic to members of this pool.
The default is **Round Robin**.
6. For the **Priority Group Activation** setting, specify how to handle priority groups:
 - Select **Disabled** to disable priority groups. This is the default option.
 - Select **Less than**, and in the **Available Members** field type the minimum number of members that must remain available in each priority group in order for traffic to remain confined to that group.
7. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
8. Click **Finished**.

The load balancing pool appears in the Pools list.

Creating a virtual server for secure electronic trading

You first need to configure a FIX profile before configuring a virtual server for electronic trading.

You can configure a virtual server for electronic trading, using a FIX profile.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.

4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. In the **Service Port** field, type the port number used for the FIX message.
6. From the **Configuration** list, select **Advanced**.
7. From the **Protocol** list, select **TCP**.
8. From the **Protocol Profile (Client)** list, select a predefined or user-defined TCP profile.
9. (Optional) For the **SSL Profile (Client)** setting, from the **Available** list, select **clientssl**, and using the Move button, move the name to the **Selected** list.
10. (Optional) For the **SSL Profile (Server)** setting, from the **Available** list, select **serverssl**, and using the Move button, move the name to the **Selected** list.
11. From the **FIX Profile** list, select the FIX profile you want to assign to the virtual server.
12. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
13. Click **Finished**.

A virtual server is configured for electronic trading, using a FIX profile.

Viewing FIX message statistics

You can view various statistics specific to FIX profile traffic.

1. On the Main tab, click **Local Traffic > Virtual Servers > Statistics**.
The Virtual Servers statistics screen opens.
2. From the **Statistics Type** list, select **Profiles Summary**.
3. In the Global Profile Statistics area, for the Profile Type **FIX**, click **View** in the Details.
The system displays information about the number of current connections, the number of messages, the total message size, and the number of messages in the last sample interval.

The FIX profile statistics are available.

Implementation result

This implementation configures a BIG-IP[®] system to manage electronic trading functionality, provides you with the ability to use Financial Information eXchange (FIX) protocol messages.

Implementing Low-Latency Electronic Trading Functionality

Overview: Configuring the BIG-IP system for low-latency electronic trading

You can configure the BIG-IP[®] system to manage traffic for low-latency electronic trading. The BIG-IP system optimizes Financial Information eXchange (FIX) protocol connections to achieve predictable latency and jitter, a critical aspect of successful low-latency electronic trading. When you acquire a special license, you can use the FastL4 profile to optimize the necessary connections, and use the Packet Velocity[™] ASIC (PVA) to minimize any latency and deliver high performance L4 throughput without software acceleration.

About induced latency for FIX connections

Induced latency, which is the latency realized after a FIX connection is established, typically has a duration of approximately 10 µsecs or less.

About using TCP protocol for FIX clients and servers

The PVA only supports the TCP protocol, which requires FIX clients and servers to establish TCP connections. When creating a virtual server to manage the traffic for low-latency electronic trading, you will want to specify the TCP protocol setting.

Task summary

There are several tasks you can perform to implement low-latency electronic trading.

Task list

Implementing low-latency electronic trading functionality

Creating a custom Fast L4 profile

Creating a pool

Creating a virtual server for low-latency electronic trading

Implementing low-latency electronic trading functionality

In order to use a BIG-IP[®] system to manage low-latency electronic trading functionality, you must first acquire a special license. Please contact your F5 Networks[®] support representative to acquire the necessary license.

You can easily configure the BIG-IP system to manage low-latency electronic trading functionality

1. Type the following command to use tmsh:

```
tmsh
```

2. Modify the bigdb.dat file to only use the Packet Velocity[™] ASIC (PVA) acceleration.

```
modify /sys db pva.acceleration value guaranteed
```

This command disables software acceleration system-wide, so that only PVA acceleration is used by the Fast L4 profile.

3. Save the `bigdb.dat` file.
`restart sys service tmm`

The BIG-IP system is configured to use only PVA acceleration.

To use the configured PVA acceleration, you must create a Fast L4 profile and associate it with a virtual server.

Creating a custom Fast L4 profile

You can create a custom Fast L4 profile to manage Layer 4 traffic more efficiently.

1. On the Main tab, click **Local Traffic > Profiles > Protocol > Fast L4**.
The Fast L4 screen opens.
2. Click **Create**.
The New Fast L4 profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. Set the **TCP Close Timeout** setting, according to the type of traffic that the virtual server will process.
6. Click **Finished**.

The custom Fast L4 profile appears in the list of Fast L4 profiles.

Creating a pool

You can create a pool of servers that you can group together to receive and process traffic.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
5. Click **Finished**.

The new pool appears in the Pools list.

Creating a virtual server for low-latency electronic trading

After you create a server pool, you need to create a virtual server that references the profile and pool you created.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.
4. For the **Destination** setting, in the **Address** field, type the IP address you want to use for the virtual server.
The IP address you type must be available and not in the loopback network.
5. From the **Configuration** list, select **Advanced**.
6. From the **Type** list, select **Performance (Layer 4)**.
7. From the **Protocol** list, select **TCP**.
8. From the **Protocol Profile (Client)** list, select a predefined or user-defined Fast L4 profile.
9. (Optional) For the **Address Translation** setting, clear the **Enabled** check box to implement direct server return (DSR) functionality.
10. (Optional) For the **Port Translation** setting, clear the **Enabled** check box.

Important: Clearing the **Enabled** check box disables network address translation (NAT) functionality. If you require NAT, you must select the **Enabled** check box.

11. In the Resources area of the screen, from the **Default Pool** list, select a pool name.
12. Click **Finished**.

The virtual server is configured to use the specified Fast L4 profile and pool.

Implementation result

This implementation configures a BIG-IP® system to manage low-latency electronic trading functionality, optimizing the system for predictable latency and jitter.

Implementing Video Quality of Experience Functionality

Overview: Video Quality of Experience profile

The BIG-IP® system's video Quality of Experience (QoE) profile enables you to assess an audience's video session or overall video experience, providing an indication of customer satisfaction. The QoE profile uses static information, such as bitrate and duration of a video, and video metadata, such as URL and content type, in monitoring video streaming. Additionally, the QoE profile monitors dynamic information, which reflects the real-time network condition.

By considering both the static video parameters and the dynamic network information, the user experience can be assessed and defined in terms of a single mean opinion score (MOS) of the video session, and a level of customer satisfaction can be derived. QoE scores are logged in the `ltm` log file, located in `/var/log`, which you can evaluate as necessary.

Task list

Creating an iRule to collect video Quality of Experience scores

Creating an iRule to collect static information about video files

Creating a video Quality of Experience profile

Creating a pool

Creating a video Quality of Experience virtual server

Creating an iRule to collect video Quality of Experience scores

You can create an iRule to use with a video Quality of Experience (QoE) profile that defines the QoE scores to collect.

1. On the Main tab, click **Local Traffic** > **iRules**.
The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.
For complete and detailed information about iRules syntax, see the F5 Networks DevCentral web site (<http://devcentral.f5.com>).
For example, the following iRule saves `Content-Type` to session DB with a 600-second lifetime.

```
...
when HTTP_REQUEST {
    set LogString "Client [IP::client_addr]:[TCP::client_port] ->
    [HTTP::host][HTTP::uri]"
    set x_playback_session_id [HTTP::header "X-Playback-Session-Id"]
}

when HTTP_RESPONSE {
    set content_type [HTTP::header "Content-Type"]
}
```

```

}

when CLIENT_CLOSED {
    catch {
        if { ($content_type contains "video") &&
            ([QOE::video available] == 1) } {
            set qoe_params [list available width height duration nominal_bitrate
                                average_bitrate freeze_period freeze_frequency mos]
            foreach param $qoe_params {
                set value [QOE::video $param]
                append params "$param=$value "
            }
            if {[string length $x_playback_session_id]}{
                log local0. "$LogString X-Playback-Session-Id:
                    $x_playback_session_id QOE::video $params"
            } else {
                log local0. "$LogString QOE::video $params"
            }
        }
    }
}

```

5. Click **Finished.**

The new iRule appears in the list of iRules on the system.

There is now an available iRule to use with a QoE profile that collects specified QoE scores.

Creating an iRule to collect static information about video files

You can create an iRule to collect static information specific to video files, primarily for use with Policy Enforcement Manager™ (PEM).

1. On the Main tab, click **Local Traffic > iRules**.
The iRule List screen opens, displaying any existing iRules.
2. Click **Create**.
The New iRule screen opens.
3. In the **Name** field, type a name between 1 and 31 characters, such as `my_iRule`.
4. In the **Definition** field, type the syntax for the iRule using Tool Command Language (Tcl) syntax.
For complete and detailed information iRules syntax, see the F5 Networks DevCentral web site (<http://devcentral.f5.com>).
For example, the following iRule collects static information specific to video files.

```

when QOE_PARSE_DONE {
    set w [QOE::video width]
    set h [QOE::video height]
    set d [QOE::video duration]
    set b [QOE::video nominal_bitrate]
    log local0. "QOE_PARSE_DONE_ENABLED: width=$w height=$h
        bitrate=$b duration=$d"
}

```

5. Click **Finished.**

The new iRule appears in the list of iRules on the system.

There is now an iRule available to collect static information specific to video files.

Creating a video Quality of Experience profile

You can use the Traffic Management shell (tmsh) to create a video Quality of Experience (QoE) profile to use with Policy Enforcement Manager™ (PEM™) or Application Acceleration Manager™ (AAM™) and determine a customer's video Quality of Experience.

1. Log in to the command-line interface of the system using the root account.
2. Open the Traffic Management Shell (tmsh).
tmsh
3. Create a video QoE profile.
create ltm profile qoe qoe_profile_name video true

This creates the video QoE profile.

Creating a pool

You can create a pool of servers that you can group together to receive and process traffic.

1. On the Main tab, click **Local Traffic > Pools**.
The Pool List screen opens.
2. Click **Create**.
The New Pool screen opens.
3. In the **Name** field, type a unique name for the pool.
4. Using the **New Members** setting, add each resource that you want to include in the pool:
 - a) Type an IP address in the **Address** field.
 - b) Type a port number in the **Service Port** field, or select a service name from the list.
 - c) To specify a priority group, type a priority number in the **Priority Group Activation** field.
 - d) Click **Add**.
5. Click **Finished**.

The new pool appears in the Pools list.

Creating a video Quality of Experience virtual server

Before creating a video Quality of Experience (QoE) virtual server, you need to have created and configured a video QoE profile.

You can assign video QoE profile to a virtual server.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. From the **HTTP Profile** list, select **http**.
4. In the Resources area, for the **iRules** setting, from the **Available** list, select the name of the iRule that you want to assign, and using the Move button, move the name into the **Enabled** list.
5. In the Resources area of the screen, from the **Default Pool** list, select a pool name.

6. Click **Finished**.
7. Log in to the command-line interface of the system using the root account.
8. Open the Traffic Management Shell (tmsh).
`tmsh`
9. Assign the video QoE profile to the virtual server.
`modify virtual_server_name profile add qoe_profile_name`

This assigns the video QoE profile and iRules to the virtual server.

Securing Client-side SMTP Traffic

Overview: Securing client-side SMTP traffic

You can add SSL encryption to SMTP traffic quickly and easily, by configuring an SMTPS profile on the BIG-IP® system. *SMTPS* is a method for securing Simple Mail Transport Protocol (SMTP) connections at the transport layer.

Normally, SMTP traffic between SMTP servers and clients is unencrypted. This creates a privacy issue because SMTP traffic often passes through routers that the servers and clients do not trust, resulting in a third party potentially changing the communications between the server and client. Also, two SMTP systems do not normally authenticate each other. A more secure SMTP server might only allow communications from other known SMTP systems, or the server might act differently with unknown systems.

To mitigate these problems, the BIG-IP system includes an SMTPS profile that you can configure. When you configure an SMTPS profile, you can activate support for the industry-standard STARTTLS extension to the SMTP protocol, by instructing the BIG-IP system to either allow, disallow, or require STARTTLS activation for SMTP traffic. The STARTTLS extension effectively upgrades a plain-text connection to an encrypted connection on the same port, instead of using a separate port for encrypted communication.

This illustration shows a basic configuration of a BIG-IP system that uses SMTPS to secure SMTP traffic between the BIG-IP system and an SMTP mail server.

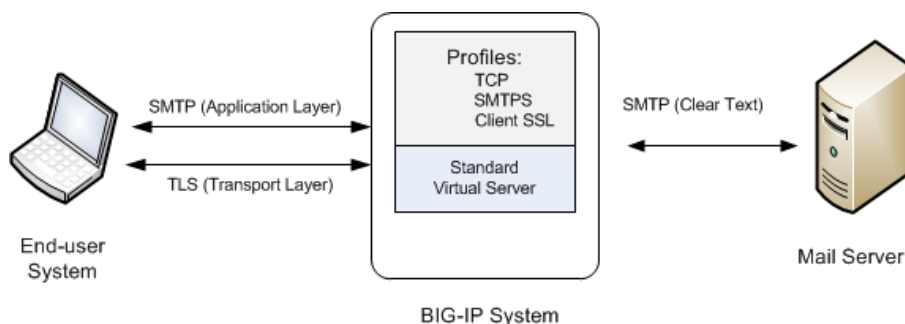


Figure 17: Sample BIG-IP configuration for SMTP traffic with STARTTLS activation

Task summary

To configure the BIG-IP® system to process Simple Mail Transport Protocol (SMTP) traffic with SSL functionality, you perform a few basic tasks.

Task list

Creating an SMTPS profile

This task specifies that STARTTLS authentication and encryption should be required for all client-side Simple Mail Transport Protocol (SMTP) traffic. When you require STARTTLS for SMTP traffic, the BIG-IP® system effectively upgrades SMTP connections to include SSL, on the same SMTP port.

1. On the Main tab, click **Local Traffic** > **Profiles** > **Services** > **SMTPS**.
The SMTPS profile list screen opens.
2. Click **Create**.
The New SMTPS Profile screen opens.
3. In the **Name** field, type a unique name for the profile.
4. Select the **Custom** check box.
5. From the **STARTTLS Activation Mode** list, select **Require**.
6. Click **Finished**.

The BIG-IP system is now required to activate STARTTLS for all client-side SMTP traffic.

Creating a Client SSL profile

You create a Client SSL profile when you want the BIG-IP® system to authenticate and decrypt/encrypt client-side application traffic.

1. On the Main tab, click **Local Traffic** > **Profiles** > **SSL** > **Client**.
The Client profile list screen opens.
2. Click **Create**.
The New Client SSL Profile screen opens.
3. Configure all profile settings as needed.
4. Click **Finished**.

After creating the Client SSL profile and assigning the profile to a virtual server, the BIG-IP system can apply SSL security to the type of application traffic for which the virtual server is configured to listen.

Creating a virtual server and load-balancing pool

You use this task to create a virtual server, as well as a default pool of Simple Mail Transport Protocol (SMTP) servers. The virtual server listens for, and applies SSL security to, client-side SMTP application traffic. The virtual server then forwards the SMTP traffic on to the specified server pool.

Note: Using this task, you assign an SMTPS profile to the virtual server instead of an SMTP profile. You must also assign a Client SSL profile.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.
The Virtual Server List screen opens.
2. Click the **Create** button.
The New Virtual Server screen opens.
3. In the **Name** field, type a unique name for the virtual server.

4. For the **Destination** setting, select the type, and type an address, or an address and mask, as appropriate for your network.
5. In the **Service Port** field, type 25 or select **SMTP** from the list.
6. From the **Configuration** list, select **Basic**.
7. For the **SSL Profile (Client)** setting, in the **Available** box, select a profile name, and using the Move button, move the name to the **Selected** box.
8. From the **SMTPS Profile** list, select the SMTPS profile that you previously created.
9. In the Resources area of the screen, for the **Default Pool** setting, click the **Create (+)** button. The New Pool screen opens.
10. In the **Name** field, type a unique name for the pool.
11. In the Resources area, for the **New Members** setting, select the type of new member you are adding, then type the appropriate information in the **Node Name**, **Address**, and **Service Port** fields, and click **Add** to add as many pool members as you need.
12. Click **Finished** to create the pool.
The screen refreshes, and reopens the New Virtual Server screen. The new pool name appears in the **Default Pool** list.
13. Click **Finished**.

After performing this task, the virtual server applies the custom SMTPS and Client SSL profiles to incoming SMTP traffic.

Implementation result

After you have created an SMTPS profile and a Client SSL profile and assigned them to a virtual server, the BIG-IP system listens for client-side SMTP traffic on port 25. The BIG-IP system then activates the STARTTLS method for that traffic, to provide SSL security on that same port, before forwarding the traffic on to the specified server pool.

Controlling Responses to ICMP Echo Requests

About ICMP echo responses on the BIG-IP system

You can control whether the BIG-IP® system sends responses to Internet Control Message Protocol (ICMP) echo requests, on a per-virtual address basis.

If you disable ICMP echo responses on a virtual address, the BIG-IP system never sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address. If you enable ICMP echo responses on a virtual address, the BIG-IP system always sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

Alternatively, you can selectively enable ICMP echo responses. Selectively enabling ICMP echo responses causes the BIG-IP system to internally enable or disable ICMP responses for the virtual address, based on which virtual server state you choose for enabling route advertisement. This table shows that for each possible virtual server state that you can specify to enable route advertisement for a virtual address, the system controls ICMP echo responses in a unique way.

Virtual server state for route advertisement	ICMP echo response behavior
When any virtual server for that virtual address is available	The BIG-IP system sends an ICMP echo response for a request sent to the virtual address, if one or more virtual servers associated with the virtual address is in an Up or Unknown state.
When all virtual servers for that virtual address are available	The BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, but only when all virtual servers are available.
When you want the system to always advertise a route to the virtual address	The BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

Task summary

You can configure the BIG-IP system to control whether the system sends a response for each ICMP echo request that is sent to a BIG-IP virtual address.

Task list

Configuring ICMP echo responses for a virtual address

You perform this task to control the way that the BIG-IP® system controls responses to ICMP echo requests sent to an individual BIG-IP virtual address. Note that the way you configure route advertisement for the virtual address can affect the way that the system controls ICMP echo responses.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen displays a list of existing virtual servers.
2. On the menu bar, click **Virtual Address List**.
3. Click the name of the virtual server you want to configure.
4. For the **ICMP Echo** setting, choose a value:
 - If you choose **Enabled**, the BIG-IP system always sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address
 - If you choose **Disabled**, the BIG-IP system never sends an ICMP echo response for an ICMP request packet sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.
 - If you choose **Selective** and route advertisement on a virtual address is set to **When any virtual server is available**, the BIG-IP system sends an ICMP echo response for a request sent to the virtual address, if one or more virtual servers associated with the virtual address is in an Up or Unknown state.
 - If you choose **Selective** and route advertisement on a virtual address is set to **When all virtual server(s) are available**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, but only when all virtual servers are available.
 - If you choose **Selective** and route advertisement on a virtual address is set to **Always**, the BIG-IP system always sends an ICMP echo response for a request sent to the virtual address, regardless of the state of any virtual servers associated with the virtual address.

Important: For those choices that depend on virtual server status, you must configure each relevant virtual server to notify the virtual address of its status.

5. Click **Update**.

After performing this task, the virtual address configuration specifies the behavior that you want the BIG-IP system to exhibit when controlling responses to ICMP echo requests.

Communicating virtual server status to a virtual address

You perform this task to control whether you want the BIG-IP® system to notify the associated virtual address of the virtual server status. The BIG-IP system uses this status to control the way that the system manages responses to any ICMP echo requests sent to the virtual address.

1. On the Main tab, click **Local Traffic > Virtual Servers**.
The Virtual Server List screen opens.
2. Click the name of the virtual server you want to modify.
3. Select or clear the **Notify Status to Virtual Address** check box.
When you select this setting, the virtual server notifies its associated virtual address of its status with respect to pool member availability.

4. Click **Update** to save the changes.

After you perform this task, the BIG-IP system reports the status of the virtual server to the associated virtual address.

Implementation results

After you complete the tasks in this implementation, the BIG-IP® system controls responses to ICMP echo requests according to the way that you configured the relevant virtual server and its virtual address.

Legal Notices

Legal notices

Publication Date

This document was published on December 28, 2017.

Publication Number

MAN-0293-10

Copyright

Copyright © 2017, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

For a current list of F5 trademarks and service marks, see <http://www.f5.com/about/guidelines-policies/trademarks/>.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: <https://f5.com/about-us/policies/patents>

Link Controller Availability

This product is not currently available in the U.S.

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and

can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Gabriel Forté.

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, <http://www.and.com>.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, *This software* refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. *Similar operating systems* includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at <http://www.perl.com>.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (<http://www.rrdtool.com/index.html>) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (<http://www.nominum.com>).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. Copyright ©1990-1994 Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. All advertising materials mentioning features or use of this software must display the following acknowledgment: This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
4. Neither the name of the University nor of the Laboratory may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes software developed by Sony Computer Science Laboratories Inc. Copyright © 1997-2003 Sony Computer Science Laboratories Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY SONY CSL AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SONY CSL OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes the GeoPoint Database developed by Quova, Inc. and its contributors.

This software incorporates JFreeChart, ©2000-2007 by Object Refinery Limited and Contributors.

This product contains software developed by the Mojarrá project. Source code for the Mojarrá software may be obtained at <https://javaserverfaces.dev.java.net/>.

This product includes software developed by McAfee®.

This product includes software developed by Ian Gulliver ©2006, which is protected under the GNU General Public License, as published by the Free Software Foundation.

This product includes the Zend Engine, freely available at <http://www.zend.com>.

This product contains software developed by NuSphere Corporation, which is protected under the GNU Lesser General Public License.

This product contains software developed by Erik Arvidsson and Emil A Eklund.

This product contains software developed by Aditus Consulting.

This product contains software developed by Dynarch.com, which is protected under the GNU Lesser General Public License, version 2.1 or above.

This product contains software developed by InfoSoft Global (P) Limited.

This product includes software written by Steffen Beyer, licensed under the Perl Artistic License and the GPL.

This product includes software written by Makamaka Hannyaharamitu ©2007-2008.

Rsync was written by Andrew Tridgell and Paul Mackerras, and is available under the GNU Public License.

This product includes Malloc library software developed by Mark Moraes. (©1988, 1989, 1993, University of Toronto).

This product includes open SSH software developed by Tatu Ylonen (ylo@cs.hut.fi), Espoo, Finland (©1995).

This product includes open SSH software developed by Niels Provos (©1999).

This product includes SSH software developed by Mindbright Technology AB, Stockholm, Sweden, www.mindbright.se, info@mindbright.se (©1998-1999).

This product includes free SSL software developed by Object Oriented Concepts, Inc., St. John's, NF, Canada, (©2000).

This product includes software developed by Object Oriented Concepts, Inc., Billerica, MA, USA (©2000).

This product includes free software developed by ImageMagick Studio LLC (©1999-2011).

This product includes software developed by Bob Withers.

This product includes software developed by Jean-Loup Gailly and Mark Adler.

This product includes software developed by Markus FXJ Oberhumer.

This product includes software developed by Guillaume Fihon.

This product includes QPDF software, developed by Jay Berkenbilt, copyright ©2005-2010, and distributed under version 2 of the OSI Artistic License (<http://www.opensource.org/licenses/artistic-license-2.0.php>).

Index

A

- adaptive connection reaping
 - configuring 164
- attacks
 - mitigating 161
- authentication
 - direct client-to-server 93
 - of clients and servers 87, 93
 - with CRLDP 167
 - with Kerberos delegation 189
- authentication constraints
 - and database proxy 122

B

- BIG-IP system
 - installing on same network 39

C

- certificates
 - creating 49, 53, 59
 - requesting from CAs 63, 67
- client and server authentication 87
- client-server authentication 93
- client-side authentication 63, 67
- Client SSL forward proxy profiles
 - creating 88
- Client SSL profiles
 - creating 50, 54, 60, 64, 69, 94, 210
- Code Red attacks
 - preventing with iRules 161
- compression profiles
 - configuring 103
- connection limits
 - calculating 165
 - to ensure system availability 157
- connection rate limits
 - about 157
 - and configuration results 157
 - creating for virtual servers 157
- connection reaping
 - configuring 164
- connection requests 157
- connections
 - creating pools for 36, 51, 61, 65, 69, 75, 83, 97, 100, 106
 - limiting 157
 - queuing TCP connection requests 155
- connection thresholds 166
- connection timers
 - setting 165
- content
 - defining with queries 23
- content adaptation 71–72, 77–78
- content adaptation configuration objects 76, 84
- content-based routing
 - about 21

- content-based routing (*continued*)
 - creating profile 22
 - viewing statistics 25
- control channel optimization 135
- cookie persistence
 - about 99
- cookie profiles
 - creating 99
- CRLDP authentication
 - configuring 167
- CRLDP configuration objects
 - creating 167
- curve name
 - specifying 59, 67
- custom FTP monitors
 - and FTP load balancing 127, 132
 - creating 127, 132
- custom monitors
 - creating 152
 - creating MS SQL 122

D

- database access
 - about user-based 122
 - and the database proxy 122
- database proxy
 - about LTM 121
- database servers
 - creating a pool 123
- data center topology
 - example of 39
- data channel optimization 135
- default route
 - for Layer 2 nPath configuration 28
 - setting 40
- Denial of Service attacks
 - filtering 161
 - mitigating 161
 - preventing 157
 - tasks for 164
 - types of 162
- destination IP addresses
 - creating for HTTP traffic 98
- DHCP lease expiration 146
- DHCP servers
 - adding as pool members 142
- DHCP virtual servers
 - implementation results 144, 146
 - overview of 145
 - overview of managing 141
 - tasks for 142, 146
- Diameter configuration
 - tasks for 193
- Diameter monitors
 - creating 193
- Diameter servers
 - monitoring 193

- Diameter service requests
 - load balancing [193](#)
- DoS attack prevention [161–162](#)
- DoS attacks, See Denial of Service attacks
- downstream nodes
 - auto-configuring [159](#)

E

- ECC (elliptic curve cryptography) [59, 67](#)
- ECDSA
 - for authentication [59, 67](#)
- ECDSA key type
 - specifying [59, 67](#)
- eCommerce traffic
 - load balancing [35](#)
- electronic trading
 - about configuring FIX profile [197](#)
 - creating virtual server for [199](#)
 - implementing with FIX profile [198](#)
 - viewing FIX message statistics [200](#)
- elliptic curve DSA
 - for authentication [59, 67](#)
- external files
 - and iRules [137](#)

F

- Fast L4 profiles
 - creating for L2 nPath routing [29, 202](#)
- files
 - importing [137–138](#)
- FIX profile
 - about configuring for electronic trading [197](#)
 - creating virtual server for trading [199](#)
 - implementing for trading [198](#)
 - viewing message statistics [200](#)
- FIX protocol
 - supported versions [197](#)
- FIX protocol connections
 - about optimization [201](#)
- FTP configuration
 - tasks for [127, 131](#)
- FTP load balancing
 - and custom FTP monitors [127, 132](#)
- FTP passive mode [127, 131](#)
- FTP profiles
 - creating [131](#)
 - defined [127](#)
- FTP traffic optimization [135](#)

H

- header values
 - for HTTP requests [72, 78](#)
 - for HTTP responses [79](#)
- health monitoring
 - described [151](#)
- health monitors
 - assigning to pools [73, 80, 89, 95, 152, 190, 199](#)
 - described [151](#)
- high-water mark thresholds [164](#)

- HTML content
 - and virtual servers [119](#)
 - modifying [115](#)
 - modifying/deleting [116](#)
- HTML tag attributes
 - modifying [115](#)
- HTTP compression
 - configuring [103](#)
 - enabling [103](#)
- HTTP compression tasks
 - off-loading from server [103](#)
- HTTP content adaptation [71–72, 77–78](#)
- HTTP profiles
 - creating [50, 54, 60, 64, 68, 75, 83](#)
- HTTP request-header values [72, 78](#)
- HTTP requests
 - adapting content for [72, 78](#)
- HTTP response-header values [79](#)
- HTTP responses
 - adapting content for [78](#)
 - compressing [103](#)
- HTTPS configuration results [52, 62, 66, 70](#)
- HTTPS traffic management
 - overview [59, 67](#)
 - overview of [49, 63](#)
- HTTP traffic
 - managing with SPDY profile [105](#)
 - using cookie persistence [99](#)
 - using source address persistence [97](#)

I

- ICAP configuration objects [76, 84](#)
- ICAP content adaptation [71, 77](#)
- ICAP profiles
 - assigning [73, 80–81](#)
- ICMP echo responses
 - and virtual server status [214](#)
 - controlling [213–215](#)
- ifile commands [137](#)
- iFiles
 - creating [138](#)
- imported files
 - listing [138](#)
- internal virtual servers
 - creating [73, 80–81](#)
- internal virtual server type
 - defined [71, 77](#)
- intranet configuration [15](#)
- IP addresses
 - checking IP reputation [46](#)
- IP address expiration [146](#)
- IP address intelligence
 - categories [46](#)
 - checking database status [46](#)
 - checking IP reputation [46](#)
 - downloading the database [43](#)
 - enabling [43](#)
 - logging information [44](#)
 - overview [43](#)
 - rejecting bad requests [45](#)
- IP intelligence database [43, 46](#)

- iprep_lookup command [46](#)
- iprep.autoupdate command [43](#)
- iprep-status command [46](#)
- IP reputation
 - overview [43](#)
- IPv4-to-IPv6 gateways
 - configuring [159](#)
- IPv6 addresses
 - load balancing to [159](#)
- IPv6 routing and solicitation messages [159](#)
- iRule commands
 - for iFiles [137](#)
- iRule events [24](#), [138–139](#)
- iRule queries [24](#)
- iRules
 - and external files [137](#)
 - and iFiles [139](#)
 - and XML routing [24](#)
 - for attack prevention [161](#)
 - for HTML content replacement [115](#)

K

- Kerberos configuration objects
 - creating [189](#)

L

- LDAP protocol [171](#), [177](#)
- load balancing
 - and monitors [151](#)
- local traffic policy
 - creating [117](#)
- logging
 - of IP address intelligence information [44](#)
- loopback interface
 - for nPath routing [30](#)
- low-latency electronic trading
 - creating virtual server for [202](#)
 - implementation overview [201](#)
 - implementing [201](#)
 - results [200](#), [203](#)
 - tasks for [197](#), [201](#)
- low-water mark thresholds [164](#)

M

- matching criteria
 - defining [22](#)
- memory utilization
 - and connection thresholds [164](#)
- monitors
 - assigning to pools [73](#), [80](#), [89](#), [95](#), [152](#), [190](#), [199](#)
 - for health checking [151](#)
 - for L3 nPath routing [32](#)
 - for performance [151](#)
- monitor types [151](#)
- MS SQL database server
 - and configuring LTM as a proxy [121](#)
- MS SQL monitor
 - creating [122](#)

- MS SQL profile
 - and statistics [125](#)
- MS SQL profiles
 - customizing to configure user-based access [123](#)

N

- namespaces
 - adding [22](#)
- network security
 - protecting [161](#)
- network topology
 - for one-IP configuration [147](#)
- Nimda worm attack
 - preventing with iRules [162](#)
- nodes
 - and connection rate limits [157](#)
- nPath routing
 - and inbound traffic [30](#)
 - and server pools [29](#)
 - configuring for L3 [31](#)
 - configuring monitors for L3 [32](#)
 - defined for L2 [27](#)
 - defined for L3 [31](#)
 - example [33](#)
 - for TCP and UDP traffic [28](#)

O

- OCSP protocol [181–182](#)
- OCSP responders
 - creating [181](#)
- OLTP
 - and virtual servers [124](#)
- OneConnect
 - creating a custom profile [124](#)
- one-IP network topology
 - illustration of [147](#)
- outgoing traffic
 - and L2 nPath routing [27](#)
 - and L3 nPath routing [31](#)

P

- packets
 - discarding [161](#)
- performance monitors
 - assigning to pools [73](#), [80](#), [89](#), [95](#), [152](#), [190](#), [199](#)
 - described [151](#)
- pool members
 - and connection rate limits [157](#)
 - restrictions on [142](#)
- pool of database servers
 - creating [123](#)
- pools
 - creating [73](#), [80](#), [89](#), [95](#), [117](#), [152](#), [190](#), [199](#), [202](#), [207](#)
 - creating for DHCP servers [142](#)
 - creating for FTP traffic [129](#), [133](#)
 - creating for HTTP [23](#)
 - creating for HTTP traffic [36](#), [51](#), [61](#), [65](#), [69](#), [75](#), [83](#), [97](#), [100](#), [106](#)
 - creating load balancing [16](#), [18](#), [159](#)

pools (*continued*)

- creating to manage Diameter traffic 194
- for HTTPS traffic 36, 55
- for HTTP traffic 148
- for L2 nPath routing 29
- for L3 nPath routing 31
- for SMTP traffic 210

profiles

- creating CRLDP 168
- creating custom Fast L4 29, 202
- creating custom SSL OCSP 182
- creating Diameter 193
- creating for client-side SSL 50, 54, 60, 64, 69, 94
- creating for client-side SSL forward proxy 88
- creating for FTP 131
- creating for HTTP 50, 54, 60, 64, 68, 75, 83
- creating for server-side SSL 94
- creating for server-side SSL forward proxy 89
- creating LDAP 172
- creating MS SQL for user-based access 123
- creating RADIUS 174
- creating Server SSL 55
- creating SSL Client Certificate LDAP 178
- creating TACACS+ 186
- creating XML 22
- for cookie persistence 99
- for FTP traffic 127, 131
- for IPIP encapsulation 31
- for L3 nPath routing 31

proxy

- about database authentication constraints 122

Proxy SSL feature

- and Server SSL forward proxy profiles 89
- and Server SSL profiles 94
- described 93
- implementing 93

R

RADIUS protocol 174

RADIUS server objects

- creating 173

rate limits 157

remote CRLDP configuration

- tasks for 167

remote Kerberos configuration

- tasks for 189

remote LDAP configuration

- tasks for 171

remote RADIUS configuration

- tasks for 173

remote server authentication

- and CRLDP protocol 167
- and Kerberos protocol 189
- and LDAP protocol 171
- and OCSP protocol 181
- and RADIUS protocol 173
- and SSL LDAP protocol 177
- and TACACS+ protocol 185

remote SSL LDAP configuration

- tasks for 177

remote SSL OCSP configuration

- tasks for 181

remote TACACS+ configuration

- tasks for 185

remote traffic authentication

- with CRLDP 167
- with Kerberos delegation 189

request-header values 72, 78

requests, excessive 157

resource consumption 161

responders

- creating for OCSP 181

response-header values 79

reverse proxy servers 113

Rewrite profile

- creating 115

Rewrite profiles

- rules for URI matching 114

route advertisement

- and ICMP echo responses 214

route domains

- and IPv6 addressing 159

routes

- defining default 149
- setting for inbound traffic 30

routing

- and XML content 21
- based on XML content 23

routing advisory messages 159

routing statistics

- for XML content 25

routing XML content 25

S

security

- for SMTP traffic 209
- of network 161

self IP addresses

- and VLAN groups 41
- creating 41
- removing from VLANs 40

self-signed certificates

- creating 49, 53, 59
- for HTTPS traffic 49, 59

server pools

- for L2 nPath routing 29
- for SMTP traffic 210

Server SSL forward proxy profiles

- creating 89

Server SSL profiles

- creating 94

SMTP security

- about 209

SMTP server pools

- creating 210

SMTPS profiles

- creating 210

SMTP traffic

- and port number 211

SNATs

- configuring client 149

- source address persistence
 - about 97
- SPDY profile
 - creating for an npn header 107
 - overview 105
- SPDY profile implementation
 - task summary 106
- SPDY traffic
 - creating virtual servers for 108
 - creating virtual servers for redirecting 107
- SSL authentication
 - configuration results 57, 96
- SSL encryption/decryption
 - configuration results 57, 96
 - with Proxy SSL feature 93
 - with SSL forward proxy feature 87
- SSL forward proxy authentication
 - configuration results 91
- SSL forward proxy encryption
 - configuration results 91
- SSL Forward Proxy feature
 - described 87
- SSL forward proxy profiles
 - creating 87
- SSL OCSP authentication 181–182
- SSL profiles
 - creating 93, 210
- SSL security
 - for SMTP traffic 210–211
- STARTTLS method
 - about 209
 - activating 210–211
- statistics
 - and MS SQL profiles 125
 - for XML routing 25
- SYN Check threshold
 - activating 166
- SYN flood attacks 161

T

- TACACS+ protocol 185
- Tcl variables 25
- TCP connection timers
 - setting 165
- TCP requests
 - queuing overview 155
- TCP traffic
 - and nPath routing 28
- timers
 - setting 165
- traffic distribution 35
- traffic forwarding
 - automating 24
- translation rules
 - for URIs 115

U

- UDP connection timers
 - setting 165

- UDP traffic
 - and nPath routing 28
- URI rules
 - requirements for specifying 114
- URI translation
 - and virtual servers 119
 - example of 113
- URI translation rules
 - 114
 - creating 115
- user-based database access
 - about 122

V

- Via header
 - disabling 112
 - identifying intermediate protocols 111
 - identifying intermediate proxies 111
 - overview 111
 - task summary 111
- video Quality of Experience
 - creating iRule to collect scores 205
 - creating iRule to collect static information 206
 - creating profile 207
 - creating virtual server 207
- Video Quality of Experience
 - overview 205
- virtual addresses
 - and loopback interface 30
 - controlling ICMP echo responses 213
- virtual server
 - creating for low-latency electronic trading 202
- virtual servers
 - and connection limits 165
 - and connection rate limits 157
 - and cookie persistence 100
 - and database transaction requests 124
 - and HTML content 119
 - and internal type 71, 77
 - and OLTP 124
 - and URI translation 119
 - applying a rate class 165
 - creating 16, 73, 80–81, 153, 160
 - creating an iRule for HTTP headers 106
 - creating connection rate limits for 157
 - creating DHCP relay type 143
 - creating for application traffic 90, 95
 - creating for Diameter traffic traffic 194
 - creating for FTP traffic 130, 134
 - creating for HTTP compression 104
 - creating for HTTPS traffic 38, 51, 56, 61, 66, 70
 - creating for HTTP traffic 37, 98, 107
 - creating for Kerberos delegation 190
 - creating for one-IP network 41
 - creating for redirecting SPDY traffic 107
 - creating for SPDY traffic 108
 - creating for video Quality of Experience 207
 - creating for web hosting 76, 84, 148
 - creating with Kerberos and SSL 191
 - DHCP relay type overview 141
 - DHCP renewal 145

- virtual servers *(continued)*
 - for DHCP renewal [146](#)
 - for inbound traffic [18](#)
 - for L2 nPath routing [27](#), [30](#)
 - for L3 nPath routing [31](#)
 - for outbound traffic [19](#)
 - for secure SMTP traffic [210](#)
 - modifying for CRLDP authentication [168](#)
 - modifying for LDAP authentication [172](#)
 - modifying for RADIUS authentication [175](#)
 - modifying for SSL Client Certificate LDAP authorization [178](#)
 - modifying for SSL OCSP authentication [183](#)
 - modifying for TACACS+ authentication [187](#)
 - setting connection limits on [166](#)
- virtual server status
 - reporting [214](#)
- VLAN external
 - creating self IP addresses for [19](#)
- VLAN groups
 - and self IP addresses [41](#)
 - creating [40](#)
- VLANs
 - enabling SNAT automap [20](#)
 - for eCommerce traffic [35](#)

- VLANs *(continued)*
 - removing self IP addresses [40](#)

W

- web servers
 - load balancing to [41](#)

X

- XML content
 - routing [21](#)
- XML content-based routing
 - and traffic forwarding [24](#)
- XML profiles
 - creating [22](#)
- XML routing
 - example of [24](#)
- XPath expressions
 - samples of syntax [23](#)
- XPath queries
 - creating [22](#)
 - rules for writing [23](#)
- XPath query
 - examples [23](#)